OPIC
OFFICE DE LA PROPRIÉTÉ INTELLECTUELLE DU CANADA

CIPO
CANADIAN INTELLECTUAL PROPERTY OFFICE

(12)(19)(CA) **Brevet-Patent**

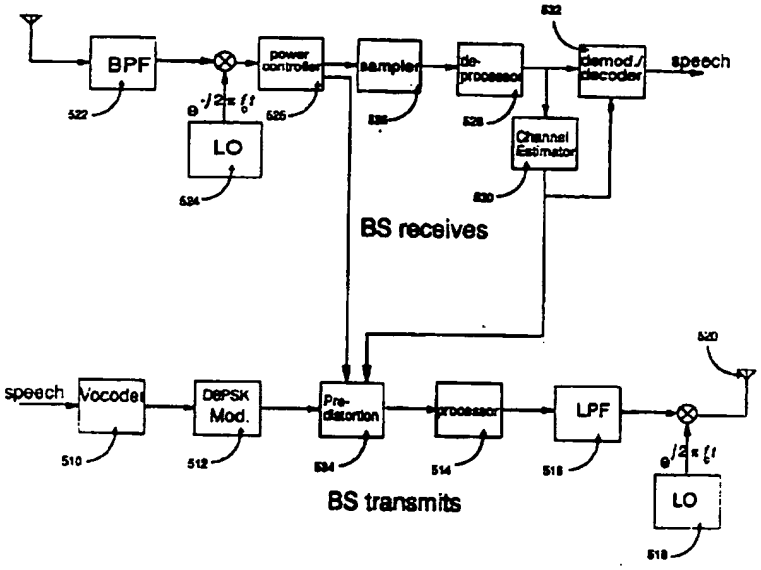
(11)(21)(C) **2,064,975**
(22) 1992/03/27
(43) 1993/09/28
(45) 1999/07/27

(72) Fattouche, Michel, CA
(72) Zagloul, Hatim, CA
(73) WI-LAN INC., CA

(51) Int.Cl.[6] H04J 1/02, H04B 1/40

(54) **METHODE ET DISPOSITIF POUR METTRE EN COMMUNICATION LES EMETTEURS-RECEPTEURS D'UN SYSTEME DE COMMUNICATION SANS FIL A ETALEMENT DU SPECTRE**

(54) **METHOD AND APPARATUS FOR MULTIPLE ACCESS BETWEEN TRANSCEIVERS IN WIRELESS COMMUNICATIONS USING OFDM SPREAD SPECTRUM**



(57) A method for allowing a number of wireless transceivers to exchange information (data, voice or video) with each other. A first frame of information is multiplexed over a number of wideband frequency bands at a first transceiver, and the information transmitted to a second transceiver. The information is received and processed at the second transceiver. The information is differentially encoded using phase shift keying. In addition, after a pre-selected time interval, the first transceiver may transmit again. During the preselected time interval, the second transceiver may exchange information with another transceiver in a time duplex fashion. The processing of the signal at the second transceiver may include estimating the phase differential of the transmitted signal and pre-distorting the transmitted signal. A transceiver includes an encoder for encoding information, a wideband frequency division multiplexer for multiplexing the information onto wideband frequency voice channels, and a local oscillator for upconverting the multiplexed information. The apparatus may include a processor for applying a Fourier transform to the multiplexed information to bring the information into the time domain for transmission.

Industrie Canada Industry Canada

BEST AVAILABLE COPY

FIELD OF THE INVENTION

This invention relates to voice and data transmission in wireless communications, and particularly between fixed and portable transmitters and receivers.

CLAIM TO COPYRIGHT

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND AND SUMMARY OF THE INVENTION

This patent document presents a new multiple access technique for Personal Communication Networks (PCN). Personal communication networks are networks that allow individuals and equipment to exchange information with each other anywhere at anytime through voice, data or video. PCN typically include a number of transceivers, each capable of transmitting and receiving information (voice, data or video) in the form of electromagnetic signals. The transceivers may be fixed or portable, and may be identical or one or more of them may be more complex.

The system must allow the transceivers to access each other to enable the exchange of information. When there are a number of transceivers, multiple access, that is, access by more than one transceiver to another transceiver, must be allowed.

One of the constraints of designing a PCN is that a transceiver, or portable radio unit, must be

small in size. The smaller the unit, the better for portability. The small size of the units means only small and light-weight power sources can be used. If the portable is to be used for any length of time, it must therefore consume minimal power.

Also, to allow use of the radio frequency spectrum without obtaining a license in North America, the system must use a spread spectrum and satisfy federal regulations. In part, these regulations impose limits on the power and the frequency spread of the signals exchanged between the transceivers. An object of an aspect of this invention is to satisfy those requirements.

Also, transceivers talk to each other over a fixed bandwidth. Because of the limited availability of the RF spectrum, the system must be bandwidth efficient yet at the same time maintain high quality exchange of information at all times in one of the most hostile channels known in communication. The new multiple access technique proposed here addresses all these issues.

The new access technique has a low Bit Error Probability (BER) as well as a low probability of dropped and blocked calls. This is due to the fact that the access technique is robust against multipath, Doppler shifts, impulse noise and narrowband interference. It has a low cochannel interference and little or no intersymbol interference.

The new access technique can offer up to 38 times the capacity of analog FM. It includes in one aspect wideband orthogonal frequency division multiplexing of the information to be exchanged, and may include slow Frequency Hopping (FH). The technique is implemented using Digital Signal Processors (DSP)

replacing conventional analog devices. The system operates with relatively small cells. In other aspects, dynamic channel allocation and voice activation may be used to improve the capacity of the system.

Advantages of the present invention include:

1. It can be used indoors as well as outdoors using the same transceivers. If data is to be exchanged, as opposed to voice, the transceiver preferably contains an estimator to allow pre-distortion and post-distortion of the transmitted signal.

2. The system, as compared with prior art systems omits the clock or carrier recovery, automatic gain control or passband limiter, power amplifier, an equalizer or an interleaver-deinterleaver, and therefore has low complexity.

3. The system offers good speech quality, as well as low probabilities of dropped and blocked calls. It is robust against Doppler and multipath shifts. It is also robust against both impulse noise and narrowband interference.

4. The system is flexible, such that at the expense of increased complexity of the DSP receiver it can be applied over noncontiguous bands. This is accomplished by dividing a 100 MHz (in one of the exemplary embodiments described here) band into several subbands each accommodating an integer number of voice channels.

5. The system offers low frame delay (less than 26.2 ms in the exemplary cellular embodiment described here). The transceiver requires low average transmitted power (of the order of 20μW in the exemplary cellular embodiment described here) which

means power saving as well as enhanced biological safety.

6. The system offers up to a 38 fold increase in capacity over the North American Advanced Mobile Phone System (AMPS) which uses analog frequency modulation.

Operation of the system in accordance with the techniques described in this disclosure may permit compliance with technical requirements for spread spectrum systems.

There is therefore disclosed in one aspect of the invention a method for allowing a number of wireless transceivers to exchange information (data, voice or video) with each other. In the method, a first frame of information is multiplexed over a number of frequency bands at a first transceiver, and the information transmitted to a second transceiver. In a cellular implementation, the second transceiver may be a base station with capacity to exchange information with several other transceivers. The information is received and processed at the second transceiver. The frequency bands are selected to occupy a wideband and are preferably contiguous, with the information being differentially encoded using phase shift keying.

A signal may then be sent from the second transceiver to the first transceiver and de-processed at the first transceiver. In addition, after a pre-selected time interval, the first transceiver transmits again. During the preselected time interval, the second transceiver may exchange information with another transceiver in a time duplex fashion.

The processing of the signal at the second transceiver may include estimating the phase

differential of the transmitted signal and pre-distorting the transmitted signal.

The time intervals used by the transceivers may be assigned so that a plurality of time intervals are made available to the first transceiver for each time interval made available to the second transceiver while the first transceiver is transmitting, and for a plurality of time intervals to be made available to the second transceiver for each time interval made available to the first transceiver otherwise. Frequencies may also be borrowed by one base station from an adjacent base station. Thus if one base station has available a first set of frequencies, and another a second set of distinct frequencies, then a portion of the frequencies in the first set may be temporarily re-assigned to the second base station.

In an implementation of the invention for a local area network, each transceiver may be made identical except for its address.

Apparatus for carrying out the method of the invention is also described here. The basic apparatus is a transceiver which will include an encoder for encoding information, a wideband frequency division multiplexer for multiplexing the information onto wideband frequency voice channels, and a local oscillator for upconverting the multiplexed information. The apparatus may include a processor for applying a Fourier transform to the multiplexed information to bring the information into the time domain for transmission.

BRIEF DESCRIPTION OF THE DRAWINGS

There will now be described a preferred embodiment of the invention, with reference to the drawings, by way of illustration, in which like numerals denote like elements and in which:

Figures 1a and 1b are schematics of a prior art receiver and transmitter respectively;

Figure 2 is a schematic showing the use of the available frequencies according to one aspect of the invention for use with cellular applications;

Figure 3a is a schematic showing an idealized pulse for transmission over a cellular system;

Figure 3b is a schematic showing a modified version of the pulse shown in Figure 3a;

Figure 3c is a schematic showing a further modified version of the pulse shown in Figure 3a;

Figure 4 is a schematic showing an exemplary protocol for cellular communication;

Figure 5a is a block diagram showing the structure and function of an embodiment of the transmitter of a cellular portable in accordance with the invention;

Figure 5b is a block diagram showing the structure and function of an embodiment of the transmitter and receiver of a cellular base station in accordance with the invention;

Figure 5c is a block diagram showing the structure and function of an embodiment of the receiver of a cellular portable in accordance with the invention;

Figure 6a is a flow diagram showing the function of the processor in either of Figures 5a or 5b;

Figure 6b is a schematic showing the function of the deprocessor in either of Figures 5b or 5c;

Figure 6c is a schematic further illustrating the operation of the processor and deprocessor shown in Figures 6a and 6b;

Figure 7a is a schematic showing the structure and function of the channel estimator in Figure 5b;

Figure 7b is a flow chart showing the operation of the channel estimator of Figures 5b and 7a;

Figures 8a, 8b and 8c are respectively schematics of 126, 63 and 7 cell reuse patterns;

Figures 9a and 9b are schematics showing transmit protocols according to one aspect of the invention;

Figure 10 is a schematic showing the use of the available frequencies according to another aspect of the invention for use with local area network applications;

Figure 11a is a schematic showing an idealized pulse for transmission over a local network system;

Figure 11b is a schematic showing a modified version of the pulse shown in Figure 11a;

Figure 11c is a schematic showing a further modified version of the pulse shown in Figure 11a;

Figure 12 is a schematic showing a preferred protocol for local area network communication;

Figure 13a is a block diagram showing the structure and function of an embodiment of the transmitter of a local area network transceiver according to the invention;

Figure 13b is a block diagram showing the structure and function of an embodiment of a further local area network transceiver according to the invention;

Figure 13c is a block diagram showing the structure and function of an embodiment of the receiver of a local area network transceiver according to the invention;

Figure 14a is a flow diagram showing the function of the processor in either of Figures 13a or 13b;

Figure 14b is a schematic showing the function of the deprocessor in either of Figures 13b or 13c;

Figure 14c is a schematic further illustrating the operation of the processor and deprocessor shown in Figures 14a and 14b; and

Figure 15 is a schematic showing the structure and function of the channel estimator in Figure 13b.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Introduction

The benefits of the invention can be readily appreciated with reference to Figure 1, which shows a prior art transmitter/receiver configuration for a portable unit. The transmitter includes a vocoder 110, an interleaver 112, a modulator 114, a filter 116, local oscillator 118, power amplifier (PA) 120 and antenna 122. The receiver includes an LNA 124, a local oscillator 126, a filter 128, automatic gain control (AGC) 130 with an associated passband hardlimiter not separately shown, carrier recovery 132, sampler 134, clock recovery 136, adaptive (or fixed) equalizer 138, demodulator 140, deinterleaver 142 and decoder 146.

With implementation of the present invention, several of the blocks shown in Figure 1 are not required. These are the interleaver 112, deinterleaver 142, power amplifier 120, automatic gain control 130 with passband hard-limiter, clock recovery 136 and carrier

recovery 132, and the equalizer 138. It will now be explained how the proposed system obtains the omission of these blocks without impairing the quality and capacity of the system.

In this disclosure there will be described two systems as examples of the implementation of the invention. The system described first here will apply to a cellular system with a number of portable transceivers and base stations (BS). Then will be described a local area network implementation. A local area network will typically be a system of equal transceivers. The invention may also be implemented with combinations of cellular and local area networks, or to a system with a number of equal transceivers and a master or controlling transceiver. "Equal" as used here means that the transceivers have more or less the same processing equipment and processing capabilities. The system described here is primarily for the exchange of voice information.

Link set-up and termination protocols between transceivers, and the equipment required to implement them, are well understood in the art as well as the basic structure of radio transceivers that may be used to implement the invention. Hence these elements are not described here. What is described here are the novel operational, functional and structural elements that constitute the invention.

Cellular Implementation of Wideband Modulation

The present invention proposes in one embodiment a wideband modulation scheme for exchange of information between transceivers such as portables and base stations.

Wideband in this patent document is described in the context of Wideband-Orthogonal Frequency Domain Modulation (W-OFDM or wideband OFDM). In OFDM, the entire available bandwidth B is divided into a number of points K, where adjacent points are separated by a frequency band Δf, that is $B = K\Delta f$. The K points are grouped into a frame of K_1 points and two tail slots of K_2 points each, so that $K = K_1 + 2K_2$. The frame carries the information intended for transmission under the form of multilevel differential phase shift keying (MDPSK) symbols or differential quadrature amplitude modulated (DQAM) symbols. Thus each point in the frame corresponds to one information symbol. The two tail slots act as guard bands to ensure that the out-of-band signal is below a certain power level. For example, when a pulse P(f) is selected for pulse shaping and the out-of-band signal has to be ydB or less relative to the in-band signal, K_2 is selected such that

$$20.\log_{10}|P(f)/P(0)| \leq y \text{ for } f \geq K_2\Delta f.$$

When the pulse is a raised-cosine pulse with a roll-off β and when the number of levels each symbol can take is M, the bit rate is equal to $K_1\log_2 M/(\delta t + (1 + \beta)/\Delta f)$ where $(1 + \beta)/\Delta f$ is the duration of the frame and δt is the guard time required to take into account the delay of arrival and the delay spread due to multipath. In this case, the bandwidth efficiency, which is defined as the ratio between the bit rate and the bandwidth, is equal to:

$$\log_2 M/((1 + \beta + \delta t\Delta f)(1 + 2K_2/K_1))$$

In wideband-OFDM, both K and Δf are selected sufficiently large to achieve a high throughput as well as to reduce the effects on the BER of the clock error, the Doppler shift and the frequency offset

between the LO in the transmitter and the one in the receiver. To show what is meant by "K and Δf are selected sufficiently large", consider the effect of increasing K and Δf on (1) the clock error, (2) the Doppler shift and (3) the frequency offset between the LO in the transmitter and the LO in the receiver.

(1) When a clock error at a transceiver of value τ occurs in the time domain, it causes a shift in the phase difference between adjacent symbols in the frequency domain of value $2\pi\Delta f\tau$. When τ is equal to χT where T is duration of one time domain sample and χ is any real value, the shift is equal to $2\pi\Delta f\chi T$. Hence, τ causes a shift in the phase difference between adjacent symbols of value $2\pi\chi/K_1$ since T is equal to $1/(K_1\Delta f)$. By doubling the number of symbols from K_1 to $2K_1$ the shift in the phase difference is reduced by half from $2\pi\chi/K_1$ to $\pi\chi/K_1$. Thus, the effect of the clock error on the BER is reduced by increasing K.

(2) When there is relative motion between the transmitting transceiver and the receiving transceiver, a Doppler shift occurs with a maximum absolute value $|V/\lambda|$ where V is the relative velocity between the two transceivers and λ is the wavelength of the travelling wave corresponding to the carrier frequency f_c (i.e. f_c is the frequency corresponding to the middle point in the frame). Such a Doppler shift causes a sampling error in the frequency domain of the same amount, or equivalently, it causes a sampling error of $V/(\lambda\Delta f)$ relative to one symbol sample. Thus, the effect of the Doppler shift on the BER is reduced by increasing Δf.

(3) When a frequency offset between the LO in the transmitter and the one in the receiver occurs

with a value f_o, it causes a sampling error in the frequency domain of the same amount, or equivalently, it causes a sampling error of $f_o/\Delta f$ relative to one symbol sample. Thus, the effect on the BER of the frequency offset between the LO in transmitter and the one in the receiver is reduced by increasing Δf.

In summary, OFDM with a K and a Δf large enough to be able to achieve a specific throughput and large enough to be able to avoid using either a clock or a carrier recovery device without substantially affecting the BER is referred to here as Wideband-OFDM. As an example, let us assume that MDPSK is used in an OFDM system with the number M of levels, with a carrier frequency f_c, with a raised cosine pulse of roll-off β, with the LO at the receiver having a frequency offset f_o relative to the LO at the transmitter (so that the frequency offset between the carrier frequencies in the first and second transceivers of the multiplexed information is f_o), with a given maximum expected clock error $\tau = \chi T$ at the receiving transceiver, where T is the duration of one time domain sample, and with a maximum expected relative velocity V between the transceivers. Thus, in order to ensure that the out-of-band signal is ydB or less relative to the in-band signal and to be able to avoid using either a clock or a carrier recovery device without substantially affecting the BER we have to:

1. Find the acceptable sampling error $\Delta f'$, relative to one symbol sample, which does not substantially affect the BER. This can be done using the following rules:

   When $0.2 \leq \beta \leq 0.3$, $\Delta f' = 7.50\%$

   When $0.3 \leq \beta \leq 0.4$, $\Delta f' = 10.0\%$

When $0.4 \leq \beta \leq 0.5$, $\Delta f' = 12.5\%$

When $0.5 \leq \beta \leq 0.6$, $\Delta f' = 15.0\%$

2. Find Δf such that:

$$V/(\lambda\Delta f) + f_o/\Delta f \leq \Delta f'$$

3. Find K_2 such that

$$20.\log_{10}|P(f)/P(0)| \leq y \text{ for } f \geq K_2\sigma f$$

4. Find K_1 such that

$$2\pi\chi/K_1 < \pi/M$$

In this case, we refer to OFDM as Wideband-OFDM. Element 4 is a necessary condition for wideband OFDM, and given a sampling error, the sampling error may be corrected with the methods described in this patent document.

To implement wideband modulation, Orthogonal Frequency Division Multiplexing (OFDM) is preferred in which the information, for example encoded speech, is multiplexed over a number of contiguous frequency bands. Wideband OFDM forces the channel to be frequency selective and causes two types of linear distortion: amplitude distortion and phase distortion. To reduce the effect of amplitude distortion the modulation is preferably phase modulation, while the effect of phase distortion is reduced by employing differential phase modulation. Hence the modulation may be referred to as Differential OFDM (DOFDM). Unlike in other proposed schemes, neither pilot tones nor diversity are required in DOFDM. Possibly, quadrature amplitude modulation might be used, but amplitude modulation makes it difficult to equalize the distorting effects of the channel on the signal.

To implement wideband modulation in a cellular system with a plurality of portables and one or more base stations, a 100MHz band is divided into 4096 points, as shown in Figure 2, plus two tail slots

of 195.3 KHz each. The 4096 points represent N voice channels (vc). Adjacent points are separated by 24.414 KHz and each point represents a Differential eight Phase Shift Keying (D8PSK) Symbol $e^{j\zeta(n)}$, where $\zeta(n) = \zeta(n-1) + \phi(n) + \chi(n)$. $\phi(n)$ takes one of the eight values $\{0, 2\pi/8, 4\pi/8, \ldots, 14\pi/8\}$ with equal probability for $n = 1, 2, \ldots, 4096$ and $\phi(0)$ takes an arbitrary value. $\chi(n)$ also takes an arbitrary value. $\chi(n)$ may be used as a security key and will be known only to the transmitter and receiver. Information in the form of output bits from a vocoder are mapped onto $\phi(n)$. Vocoders are well known in the art and do not need to be described in detail here. The focus here is to transmit the bits with an acceptable Bit Error Rate, i.e. with a BER $\leq 10^{-2}$ for voice and $\leq 10^{-8}$ for data.

Ideally, 40.96 μs (=1/24.414 KHz) is the minimum duration required for one frame to be transmitted without frequency domain intersymbol interference. This can be achieved using a Raised Cosine (RC) pulse with zero roll-off, as shown in Figure 3a. Figure 3a illustrates a rectangular (time domain) window corresponding to the RC (frequency domain) pulse. Such a pulse, however, requires an infinite frequency band. To alleviate such a requirement, an RC pulse with a 20% roll-off (i.e. $\beta = 0.2$) may be used as shown in Figure 3b. The frame duration has increased by 20% to 49.152 μs. The two tail slots of 195.3 KHz each (i.e. 8 points each) ensure that the signal outside the entire band of 100.39 MHz is below -50 dB. To allow the frame to spread over the time as a consequence of the multipath nature of the channel, an excess frame duration of

2.848 μs is provided as shown in Figure 3c, making the frame duration 52 μs in total.

Since the frame duration is 52 μs, the frame rate is 252 frames per 13.104 ms or equivalently, 126 full duplex frames may be transmitted/received every 13.104 ms. The reason for pre-selecting an interval of 13.104 ms is to ensure a transmission delay to allow one transceiver to communicate with other transceivers at the same time, but must not be so long that the delay becomes unacceptable to the user. Delays longer than about 40 ms are too great for voice, and it is preferable to be lower. For data, the delay may be longer and still be acceptable.

In the exemplary embodiment described here, three bit rates are considered for the vocoder: 18.77 Kbps, 9.16 Kbps and 6.18 Kbps. Table I displays the structure of a vc slot and the number N of vc for each vocoder rate. The control symbols in each vc slot are required for handoff and power control. Figure 2 shows that N vc can be transmitted simultaneously. This is known as Frequency Division Multiple Access. Figure 3c shows that 126 full duplex frames can be transmitted every 13.104 ms in a Time Division Multiple Access fashion (TDMA). The total number of Full Duplex voice channels (FDvc) is therefore 126xN and is shown in Table I.

To ensure that the channel is slowly fading, a Time Division Duplex protocol for exchange of information between the portable and the base station is proposed as illustrated in Figure 4. The protocol is as follows:

1. The portable transmits a frame 410 over one vc slot. See the discussion in relation to Figure 5a below.

2. The Base Station (BS) receives the frame 410 from the portable and processes (analyzes) it as shown and discussed in relation to Figure 5b below.

3. Based on the received signal, the BS predistorts a frame 420 and transmits it to the portable over the same vc slot, 520 μs or some other suitable time interval later in which the channel does not change substantially. The time interval will depend on factors such as the frequency, speed of the transceiver and other environmental factors.

4. The portable receives the frame from the BS. See the discussion in relation to Figure 5c below.

5. Steps 1 through 4 are repeated, as for example by the transmission of the next frame 430, every 13.104 ms until the call is terminated.

During 520 μs, a portable travelling outdoor at 100 km/hr moves 1.44 cm, which leaves the outdoor channel largely unchanged. Indoors, a portable moving at 2 m/s moves 0.1 cm again leaving the channel unchanged. Assuming that the channel is reciprocal and stationary over 520 μs, a predistorted signal, transmitted by the BS, should reach the portable undistorted.

From Figure 4, one can see that the portable transmits/receives one FDvc every 13.104 ms, while the BS can transmit/receive up to 21 frames or equivalently up to 21xN FDvc every 13.104 ms. The frames 440 labelled frame 2...frame 21 are frames that may be transmitted to other portables. This implies that while one BS is processing its data over 520 μs, six other BS can communicate to their corresponding portables in a Time Division Multiple Access (TDMA) fashion using the same frequency bands. Also, during the 13.104 ms, or such other preselected time interval

that is suitable, the BS may communicate with one or more other portables.

When a portable is stationary during a call, it is possible with high probability to have the transmitted signal centered with several deep (frequency domain) nulls, hence, causing speech degradation. Also, narrowband interference over the vc slot can deteriorate the speech. In order to avoid both situations, the signal is preferably frequency hopped into a new vc slot within the same (frequency domain) frame. This frequency hopping is ordered by the BS which is constantly monitoring the channel frequency response. Monitoring techniques, as well as frequency hopping, are known in the art, and not described here further. When an unacceptable speech degradation is first noticed by the BS a probation period is initiated and maintained for at least 10 cycles (i.e. 10x13.104 ms) unless speech degradation has ceased. In other words, the probation period is terminated if speech degradation has ceased. Frequency hopping is then ordered at the end of the probation period. The period of 10 cycles is long enough to indicate the portable stationarity and is short enough to allow speech interpolation between unacceptable speech frames, hence maintaining good speech quality. As known in the art, the BS ensures that no collisions take place between hopping portables.

Digital Signal Processing

The transmitter/receiver block diagrams corresponding to the protocol in Figure 4 are shown in Figures 5a, 5b and 5c. Figure 5a corresponds to step 1 in the protocol described above. Speech is provided to a vocoder 510 where the speech is digitized and

coded to create bits of information. The bits are provided to the modulator 512 which turns them into D8PSK symbols, with three bits per symbol. The D8PSK symbols are then processed in the processor 514 which is described in more detail in Figure 6a. The output from the processor is then filtered in low pass filter 516, upconverted to RF frequencies using local oscillator 518 and transmitted by antenna 520. Figure 5b corresponds to steps 2 and 3.

In Figure 5b, the received signal at the base station is filtered in a bandpass filter 522, and down converted by mixing with the output of a local oscillator 524. The average power of the downcoverted signal is monitored by a power controller 525 that adjusts the average power to the specifications required by the sampler 526. The adjusted downconverted bits are then sampled in sampler 526 to produce bits of information. The bits are then processed in the deprocessor 528, described in more detail in Figure 6b. An estimate of the phase differential of the received signal is taken in the channel estimator 530, as described in more detail in relation to Figure 7a and 7b below, and the estimated phase differential is supplied to a decoder-demodulator 532 to correct the received bits. The estimated phase differential is also supplied to a pre-distorter 534 in the transmitter. At the transmitter in the Base Station, the same blocks are incorporated as in the portable transmitter except that a pre-distorter is used to alter the phase of the D8PSK symbols to make the channel appear Gaussian (ideal) as opposed to a fading channel. The pre-distorter 534 receives a signal corresponding to the estimated phase differential of the channel. On the

(believed reasonable) assumption that the channel is reciprocal, the signal being transmitted is pre-distorted with the estimated phase differential so that the received signal at the portable with which the BS is communicating will be corrected for any phase distortion over the channel. The advantage of rendering the channel Gaussian is a large saving in the power required to achieve an acceptable BER. The initial power control 525 also sends a signal to the pre-distorter 534 to adjust the transmitted power to an appropriate signal level for the sampler 526 in the portable's receiver depending on the average power of the received signal. Thus if the average power is too low, the transmitted power is increased and if the average power is too high, the transmitted power is decreased. The power controller 25 may also be used in frequency hopping to monitor the average power of the received frequency and determine when frequency hopping need take place.

Figure 5c corresponds to step 4, and shows the receiver of the portable, which is the same as the receiver in the BS except it does not include an estimator or a power controller. These are not required in the portable on the assumption that the BS will carry out the phase estimation and the power control. However, if desired, the portable may include these functions.

Figures 6a, 6b and 6c illustrate the function and structure of the processor and the deprocessor respectively in the transmitter and receiver. Software for modelling the function of the processor in a general purpose computer is attached hereto as Appendix A, and for modelling the function of the deprocessor as Appendix B.

Figure 6a shows that the processor is a DSP implementation of an RC pulse shaping filter with a 20% roll-off, followed by an inverse Fourier transform. The processor first inverse Fourier transforms the 4096 D8PSK modulated symbols output from the modulator. The transformed symbols are then triplicated as a group so that the total number of samples is tripled, with three consecutive groups each consisting of the 4096 transformed symbols. The triplication of the signal is illustrated in Figure 6c, where the symbols are shown as first delayed and added together. Next, as shown in Figures 6a and 6c, the three groups are windowed by a Raised Cosine window with a roll-off of 0.2 centered in the middle of the three groups. In other words, the processor takes D8PSK symbols in, pulse shapes them and inverse Fourier transforms them. On the other hand, the deprocessor undoes what the processor did, i.e. it removes the pulse shaping, then Fourier transforms the received signal to obtain the original D8PSK symbols. The first two blocks in Figure 6b are similar to the second two blocks in Figure 6a except for two differences. The two differences are as follows. In the first block of the deprocessor, the repeated groups of symbols are partially overlapped as shown in Figure 6c on the right hand side. In the second block, a rectangular window is used instead of the Raised Cosine. In the preferred implementation, the blocks are repeated three times but other numbers of repetition may be used.

Figures 6a, 6b and 6c show that the DSP blocks used in the processor are identical to the ones used in the deprocessor, except for a small change in the two transforms and a small change in the shapes of

the two windows. Thus the same hardware can be used by both the processor and the deprocessor.

Figure 7a shows a block diagram of an example of a preferred channel estimator, and Figure 7b is a flow chart showing the operation of the phase estimator. Each of the steps is carried out in a computing means that may be a special purpose computer or a general purpose computer programmed to carry out the digital signal processing described here, as for example with the software that is attached hereto as appendix C. Other methods of estimating the channel may be used that obtain an estimate of the channel group delay or phase differential of the transmitted symbols. However, a preferred implementation is described here.

The first block in Figure 7a estimates the envelope A(n) for n = 1, ..., 4096 of the (frequency domain) samples transmitted over the fading channel as output from the deprocessor. The estimate A'(n) is the square-root of the sum of the squares of the quadrature (Q) and inphase (I) samples output from the deprocessor. The second block performs the operation: $\Delta\ln(A'(t)) = (A'(n) - A'(n-1))/A'(n)$, for n = 2, ..., 4096, where A'(n) is the estimate of A(n). The third block performs a Hilbert transform operation $H[\Delta\ln(A'(t))]$ on the result of the second block. $H[\Delta\ln(A'(t))]$ is an estimate of $|\Delta\omega(n)|$ for n = 2, ..., 4096, where $\Delta\omega(n)$ is the phase differential of the transmitted signal (ω is the phase of the signal). The Hilbert transform is preferably carried out by taking the discrete fast Fourier transform of the data record, multiplying the positive frequency spectrum of the transform by -i (square root -1), and the negative frequency spectrum of the transform by i, and taking

the inverse discrete fast Fourier transform. The result is a set of symbols representing an estimate of the phase differential of the received signal, as determined from its sampled amplitude envelope.

Instead of a Hilbert transform, a different estimation may be made to estimate the phase differential. In this case, firstly, after the electromagnetic signal has been sampled, a series of data frames of a number of consecutive amplitude samples (A(t)) of the electromagnetic signal are constructed. These data frames are then segmented into segments $[t_1, t_2]$, where the amplitude of the electromagnetic signal is at least a predetermined number of dB less than its running mean, for example, 10dB. The following calculation is then applied to these segments of the amplitude samples:

$$\Delta\omega(t) \approx 1/t_0 \frac{-1}{1 + (t'/t_0)^2}$$

where $t' = t - t_{min}$, t_{min} is the time in $[t_1, t_2]$ when A(t) reaches its minimum, t is the time from the beginning of the segment, and t_0 is the time from the instant the amplitude of the electromagnetic signal reaches its minimum during the segment until the amplitude reaches double its minimum during the segment. In other words, the phase differential may be calculated from

$$\Delta\omega(t) \approx -t_0/(t_0^2 + t'^2).$$

The polarity of $\Delta\omega(n)$ is extracted using the last block shown in Figure 7a. The estimate so calculated does not provide the sign of the differential. This may be determined by known techniques, for example by adding the phase differential to and subtracting the phase differential

from the received phase ($\tan^{-1}$ (Q/I)) and taking the sign to be positive if the addition results in the smaller Euclidean distance to the expected value and negative if the subtraction results in the smaller Euclidean distance to the expected value.

Equivalently, for each sample n, the ideal phase closest to $\omega(n)+\Delta\omega(n)$ is determined and labelled $\omega_+(n)$, and the ideal phase closest to $\omega(n)-\Delta\omega(n)$ is determined and labelled $\omega_-(n)$. The two sums $P = \Sigma|\omega_+(n) - \{\omega(n)+\Delta\omega(n)\}|$ and $N = \Sigma|\omega_+ - \{\omega(n)-\Delta\omega(n)\}|$ are calculated. If $P < N$, then $\omega(n)+\Delta\omega(n)$ is used to correct the signal, and if not then $\omega(n)-\Delta\omega(n)$ is used to correct the signal.

For simplicity of the estimator, the determination of the sign need only be carried out for phase differentials greater than a predetermined threshold. This will be in the vicinity of a fade and may be accomplished by segmenting the data record into a segment in which the phase differential is larger than a selected threshold and setting the remainder of the data record to zero. This computation may be carried out with a simple discrimination circuit or equivalent computing means in the estimator.

The bias $\delta\omega$ of the channel group delay is estimated by averaging $\Delta\omega'(n)$ over n for $n = 1, \ldots, 4096$ where $\Delta\omega'(n)$ is the measured value of $\Delta\omega(n)$. The estimates $A'(n)$ and $\Delta\omega'(n)$ are used directly in the predistortion filter in Figure 5b, while the estimates $\Delta\omega(n)$ and $\delta\omega$ of the unbiased channel group delay and of the bias of the channel group delay respectively are used in the demodulator.

The complexity of the processor-deprocessor-channel estimator is displayed in Table II. Complexity is measured in Mega Instructions Per Second (MIPS)

where one instruction is defined as one complex addition, one complex multiplication and a storage of one complex number. It does not include overhead.

The complexity of the processor-deprocessor-channel estimator in the BS is computed from the complexity of the Inverse Fast Fourier Transform (IFFT)/Fast Fourier Transform (FFT)/Hilbert Transform. The complexity is 4096 x 12 x 4 x 21/13.104ms for the BS. For the portable, it is computed from the complexity of the FFT/IFFT per vc: (32 x 5 + 64 + 128 + 256 + 512 + 1024 + 2048 + 4096)2/13.104ms for the portable with a 6.18 Kbps vocoder. Such a complexity assumes that the A/D converter operates at 100 MHz with 12 bit precision. As seen in Table II, the portable has smaller complexity due to the fact that the portable transmits/receives one vc in 13.104 ms and the BS transmits/receives up to 21xN vc in 13.104 ms.

Reducing Analog Complexity

Comparing Figure 1 (prior art) and Figure 5, it will be seen that several conventional blocks are not used in the present invention, namely the interleaver-deinterleaver, the Power Amplifier (PA), both the clock and the carrier recovery, both the AGC with its associated Passband hard limiter, as well as the equalizer.

From the BS point of view, the interleaver-deinterleaver is not required since the signal is predistorted before transmission forcing the received samples to be independent. From the portable point of view, the interleaver-deinterleaver is not required as a separate entity from the vocoder due to the fact that the channel is highly frequency selective, hence

the interleaving/deinterleaving can be applied implicitly in the vocoder over one vc, without a need for a separate time domain interleaver/deinterleaver. This eliminates excess speech delays associated with interleaving/deinterleaving between frames.

The PA is not required since the cells can have, as shown later, a radius of up to at least 250 m outdoors and 30 m indoors, if the transmitted power is up to 6 dBm. Such a power can be generated by the Local Oscillator (LO) without a need for a PA. It is important to avoid using a PA since DOFDM generates a time domain signal with non constant envelope. A power efficient class C PA cannot be used without distorting the signal. A class A PA can be used at the expense of power efficiency.

A clock recovery device is not required since a sampling error in the time domain is equivalent to a phase shift in the frequency domain. The phase shift is a linear function of frequency. It contributes to the bias in the channel group delay. Such a bias can be easily estimated and removed as mentioned previously by averaging $\omega'(n)$ over n in the frequency domain. Such an estimate is accurate as long as the sampling error is less than 0.2 μs or equivalently less than 20 samples (since in this case, the corresponding phase shift is less than π), and as long as the number of points in one vc is large enough as it is here.

A carrier recovery device is not required since a carrier offset in the time domain is equivalent to a sampling error in the frequency domain. For the chosen RC pulse, a sampling error of up to 10% of the duration of one pulse is acceptable. This implies that a frequency offset of up to 2.414

KHz is acceptable regardless whether it is due to carrier offset as low as 1 part in a million, i.e. as low as 1 KHz per 1 GHz. When a carrier frequency higher than 2.414 GHz is required, one can decrease in Figure 2 the number of points per 100 MHz or one can use an RC pulse with a rolloff larger than 20%.

Neither an AGC nor a Passband hard-limiter are required since the level of the received power may be controlled constantly. This is achieved as follows: The portable transmits a frame. The BS receives the frame and predistorts a frame intended for transmission accordingly, assuming that the channel is reciprocal and stationary over 520 μs. This includes controlling the transmitted power according to the received power. The BS transmits the predistorted frame and simultaneously orders the portable to control its power. The order is conveyed using the control symbol in the vc slot (See table I). The degree of power control may be determined using the power controller 525, and the instruction for the inclusion of a power control symbol in the vc may be sent from the power controller 525 to the predistorter 534.

One advantage of wideband modulation over narrowband modulation is that the wideband signal does not experience short term fading the same way the narrowband one does. The wideband signal is mainly affected by shadowing and other long term effects which vary slowly and are easily monitored from one frame to the other as long as the same vc slot is used by the portable to transmit and receive (i.e. as long as TDD is employed).

Finally, conventional equalization, whether it is linear or nonlinear, is not required simply

because there is little or no ISI. Also, from the portable point of view, each received vc is predistorted by the BS. Hence, the channel can be modeled approximately as an ideal memoryless Additive White Noise Gaussian (AWGN) channel, assuming channel reciprocity and stationarity over 520 μs. From the BS point of view, since the received signal is not predistorted by the portable prior transmission, the channel estimator is used to reduce the effect of the channel group delay.

Smaller cells

As mentioned previously, the LO generates a 6 dBm average power, hence the signal power transmitted by the BS over one vc slot is (6 dBm - $10\log_{10}N$ dB) while the signal power transmitted by the portable over one vc slot is 0 dBm. Also, since the noise power over a 100 MHz band is -94 dBm, it is (-94 dBm - $10\log_{10}N$ dB) over one vc. A typical noise figure at the receiver is 7 dB. The penalty for not using a matched filter in the receiver is 1 dB. Combining together the above figures provides the portable with an (86 dB - path loss in dB) received signal to noise ratio (SNR), while it provides the BS with an (86 dB + $10\log_{10}N$ dB - path loss in dB) received SNR.

For a path loss of 75 dB, the radius of the urban cell can be 250 m while it can be 30 m for the indoor cell. Such a path loss provides the portable with a 17 dB received SNR, while it provides the BS with an (11 dB + $10\log_{10}N$ dB) received SNR. From the portable point of view, the channel can be modeled approximately as an ideal AWGN channel, hence the 17 dB received SNR results in a 2×10^{-3} BER. On the other hand, the channel can be pessimistically modeled as a

Rayleigh fading channel from the BS point of view. The corresponding BER are displayed in Table III which shows that the achieved BER is $\leq 4x10^{-3}$. A BER $\leq 10^{-2}$ is acceptable for voice.

Cell Pattern Reuse

From Table I, the number of Full Duplex voice channels (FDvc) that can be transmitted/received per frame is 136 over 100 MHz, for a 6.18 Kbps vocoder. If the bandwidth is halved to 50 MHz, the number of FDvc per frame is reduced to 68, the noise floor is reduced by 3 dB and the number of full duplex frames that a BS can transmit/receive is doubled to 42, leaving the frame duration, the number of frames per 13.104 ms and the processor/deprocessor complexity unchanged.

Reducing the available bandwidth directly affects the cell pattern reuse. This can be explained as follows, assuming that we are required to offer a minimum of 136 FDvc per cell, that the vocoder rate is 6.18 Kbps and that the cell radius is fixed at 1 km outdoors and 30 m indoors. For a 100 MHz band, we assign one frame per cell and offer 136 FDvc per cell. In this case, the cell pattern reuse consists of 126 cells as shown in Figure 8a which displays a seven layer structure. For a 50 MHz band, we assign two frames per cell and offer 136 FDvc per cell, hence reducing our cell pattern reuse to a 63 cell pattern as shown in Figure 8b which displays a five layer structure. If the available bandwidth is as low as 5.86 MHz, we have 8 vc per frame. Hence we have to assign 18 frames per cell in order to offer the minimum required number of FDvc per cell. This reduces the cell pattern reuse to as low as a 7 cell pattern

as shown in Figure 8c which displays a two layer structure.

In Figures 8a, b and c, a shaded area is shown around the center of the pattern, indicating 19, 38 and 126 full duplex frames that the central BS can transmit/receive respectively. Tables IVa, b and c show the number of cell layers in each cell pattern reuse, the coverage area in Km^2 of the pattern reuse for both the indoor and the urban environments, as well as the carrier to interference ratio (CIR) in dB, for the 100 MHz, 50 MHz and 5.96 MHz bands, respectively. In all cases, the CIR is large enough to sustain a toll quality speech.

Transmission/Reception Protocol

Since the number of FDvc a portable can transmit/receive is one, while the number of FDvc a BS can transmit/receive is much larger as shown in Table V for each of the three vocoder rates, we have chosen the following transmission/reception protocol:

1. The portable transmits a frame over a vc.
2. Seven adjacent BS receive the frame from the portable.
3. One BS transmits to the portable, depending for example on the strength of the received signal by each of the BS.

The control of this protocol may use any of several known techniques. For example, the commonly used technique is to have the portable monitor the channel and determine which of several base stations it is closest to. It can then order the nearest BS to communicate with it. Another technique is to use a master control which receives information about the strength of the signal on the channel used by the

portable and controls the BS accordingly. Such techniques in themselves are known and do not form part of the invention.

Such a protocol has several advantages. For instance, the location of the portable can be determined with high accuracy based on the received vc at the seven adjacent BS. Locating the portable can assist in the BS hand-off. A BS hand-off and a portable hand-off do not necessarily occur simultaneously, contrary to other prior art systems. In the present invention, when a portable roams from one cell X to an adjacent cell Y, a new vc is not required immediately. What is required is a BS hand-off, meaning that BS Y (associated with cell Y) must initiate transmission to the portable over the same vc, while the BS X (associated with cell X) must terminate its transmission to the portable.

A BS hand-off occurs without the knowledge of the portable and can occur several times before a portable hand-off is required. A portable hand-off is required only when the CIR is below a certain level. In this case, the Mobile Telephone Switching Office (not shown) calls for a portable hand-off in accordance with known procedures. Reducing the portable hand-off rate reduces the probability of dropped calls. This is because a dropped call occurs either because the portable hand-off is not successful or because there are no available channels in cell Y.

The present invention allows the use of post-detection diversity at the BS, and the use of dynamic channel allocation (DCA).

Dynamic Channel Allocation

DCA is made possible by each BS having capability to transmit/receive more than the number of FDvc allocated to its cell, namely seven times the number of FDvc for a 5.86 MHz band and up to twenty-one times the number of FDvc for a 100 MHz as well as a 50 MHz band. The DCA protocol simply consists of borrowing as many FDvc as needed from the adjacent cells, up to a certain limit. The limit for the case when we employ a 6.18 Kbps vocoder, a 5.86 MHz band and 18 frames per cell is obtained as follows. The cell reuse pattern consists of 7 cells. Each cell is preassigned 144 FDvc. Assuming that at peak hours, 75 FDvc are used on the average and 5 FDvc are reserved at all times, then we are left with 64 idle channels which represent the limit on the number of FDvc one can borrow from the cell.

One should distinguish between the limit on the channels borrowed and the limit on the nonpreassigned channels a BS can use. For instance, if a call originates in cell X and the portable roams into an adjacent cell Y where no preassigned cells are available, BS Y does not need to borrow immediately a new channel from an adjacent cell. It can use the original channel as long as the level of CIR is acceptable. If on the other hand, a portable wants to initiate a call in cell Y where all preassigned channels are used, BS Y can borrow a channel from an adjacent cell up to a limit of 64 channels per cell.

The main advantage of DCA over Fixed Channel Allocation (FCA) is the increase in traffic handling capability. For FCA, a 7 cell pattern each with a preassigned 144 Fdvc can carry a total traffic of 880.81 Erlang at 0.01 Blocking Probability (BP). For

DCA, a 7 cell pattern consists of 6 cells each with 80 FDvc that can carry a total traffic of 392.17 Erlang, combined with one cell with 528 FDvc that can carry 501.74 Erlang. The total traffic is therefore 893.91 Erlang. This increase appears to be marginal (1.5%). However, if 501.74 Erlang are actually offered to one cell in the FCA system (with 14 FDvc/cell), while the six other cells carry 392.17/6 = 65.36 Erlang per cell, the BP at that busy cell 0.714 while it is negligible at the six other cells. The total blocked traffic (i.e. lost traffic) in the FCA system is then equal to (6x65.36x0.0 + 1x0.714x501.24) = 358.24 Erlang. This represents a 0.4 average BP. If the DCA is allowed such a loss, its traffic handling capacity would increase to 1768.04 Erlang which represents a 100% increase in traffic handling capacity over the FCA system, or equivalently a 160% increase in the number of available FDvc. The DCA system thus represents a marked improvement over the FCA system.

Voice Activation

Voice activation is controlled by the BS according to techniques known in the art. At any instant during a conversation between a BS and a portable, there are four possibilities:

1. BS talks while the portable listens.
2. BS listens while the portable talks.
3. BS and portable talk simultaneously.
4. BS and portable listen simultaneously.

The BS controls the voice activation procedure by allocating in cases 1, 3 and 4 three slots (frames 1.1, 1.2 and 1.3) to the BS and one slot the portable (frame 1) every four slots as shown in

Figure 9a. Likewise up to 21 portables may communicate with the base station in like fashion.

In case 2, on receiving a signal from the portable, the BS allocates three slots (frames 1.1, 1.2 and 1.3) to the portable and one slot (frame 1) to the BS every four slots as shown in Figure 9b. Likewise, up to 21 other portables may communicate with the base station in like fashion. Consequently, instead of transmitting two full duplex voice frames over four slots as in Figure 4, voice activation allows us to transmit three full duplex voice frames over four slots. Hence, voice activation provides a 50% increase in the number of available FDvc at the expense of increasing DSP complexity.

Capacity

The capacity of Code Division Multiple Access (CDMA) may be defined as the number of half duplex voice channels (HDvc) effectively available over a 1.25 MHz band per cell. Based on such a definition, Table IV displays the capacity of analog FM and of the present system with a 6.18 Kbps vocoder, 5.86 MHz band, 1 frame per cell and DCA. As shown in Table IV, the capacity of analog FM is 6 HDvc/1.25MHz/cell while for the present system it is 150 HDvc/1.25MHz/cell. The 6.25 MHz band consists of 5.86 MHz plus two tail slots. When voice activation is used, the capacity of the present system is increased by 1.5 times to 225 HDvc/1.25MHz/cell, a 38 fold increase over analog FM.

Local Area Networks

The invention may also be applied to produce a 48 Mbps wireless LAN, which also satisfies the technical requirements for spread spectrum.

For wireless LAN, wideband differential orthogonal frequency division multiplexing is again employed. The LAN will incorporate a plurality of transceivers, all more or less equal in terms of processing complexity, and possibly with identical components except for addresses.

To implement wideband modulation for a LAN, a 26 MHz band is divided into 128 points, as shown in Figure 10, plus two tail slots of 1.48 MHz each within the 26 MHz band. Adjacent points are separated by 180 KHz and each point, as with the application described above for a portable-base station, represents a D8PSK symbol. The transmitter components will be the same as shown in Figure 5b, with suitable modifications as described in the following, and will include an encoder. The output bits from the encoder are mapped onto the D8PSK symbols.

The frame duration for the symbols is illustrated in Figure 11. A rectangular time domain window corresponding to a RC frequency domain pulse has a 5.55 μs duration, and includes a 25% roll-off and excess frame duration of 0.26 μs, making a total 7.2 μs duration for the frame.

For such a wireless local area network (LAN), in which the transceivers are equal, the Time Division Duplex protocol is as illustrated in Figure 12 (assuming there are at least a pair of transceivers):

1. A first transceiver transmits a signal (frame 0) over the entire frame.

2. A second transceiver receives the signal from the first transceiver and processes (analyzes) it.

3. Based on the received signal, the second transceiver predistorts and transmits nine frames (frames 1 - 9) to the first transceiver immediately.

Each transceiver has transmitter components similar to those illustrated in Figure 5b, with suitable modifications to the internal structure to allow the use of the particular frequency band and frame duration employed.

The transmitter/receiver functional and structural block diagrams are shown in Figures 13a, 13b and 13c for the exchange of data. Data is provided to an encoder 810 where the speech is digitized and coded to create bits of information. The bits are provided to the modulator 812 which turns them into D8PSK symbols, with three bits per symbol. The D8PSK symbols are then processed in the processor 814 which is described in more detail in Figure 14a. The output from the processor is then filtered in low pass filter 816, upconverted to RF frequencies using local oscillator 818 and transmitted by antenna 820.

In Figure 13b, the received signal at the base station is filtered in a bandpass filter 822, and down converted by mixing with the output of a local oscillator 824. The average power of the downcoverted signal is monitored by an initial power control 825 that adjusts the average power to the specifications required by the sampler 826. The adjusted downconverted signal is then sampled in sampler 826 to produce bits of information. The bits are then processed in the deprocessor 828, described in more detail in Figure 14b. An estimate of the phase differential is taken in the channel estimator 830, as

described in more detail in relation to Figure 7 above, and the estimated phase differential is supplied to a decoder/demodulator 832 to correct the received bits. The estimated phase differential is also supplied to a pre-distorter 834 in the transmitter. At the transmitter in the Base Station, the same blocks are incorporated as in the portable transmitter except that a pre-distorter is used to alter the phase of the D8PSK symbols to make the channel appear Gaussian (ideal) as opposed to a fading channel. The initial power control 825 also sends a signal to the pre-distorter 834 to adjust the transmitted power to an appropriate signal level for the sampler 826 in the first transceiver. It will be appreciated that a pre-distorter will be included in the first transceiver's transmitter but that it will not be operable, except when the first transceiver is operating as a base station.

Figure 13c shows the functional blocks of the receiver of the first transceiver, which is the same as the receiver in the second transceiver except it does not include an estimator. The processor is illustrated in Figure 14a and 14c and the deprocessor in Figure 14b and 14c. The processor first inverse Fourier transforms the 128 D8PSK symbols output from the modulator. The transformed symbols are then triplicated as a group so that the total number of samples is tripled (see the left side of Figure 4c), with three consecutive groups each consisting of the 128 transformed symbols. Next, the three groups are windowed by a Raised Cosine window with a roll-off of 0.25 centered in the middle of the three groups. In other words, the processor takes D8PSK symbols in, pulse shapes them and inverse Fourier transforms them.

On the other hand, the deprocessor undoes what the processor did, i.e. it removes the pulse shaping, then Fourier transforms the received signal to obtain the original D8PSK symbols. The first two blocks in Figure 14b are similar to the second two blocks in Figure 14a except for two differences as follows. In the first block shown in Figure 14b, the repeated groups of symbols are partially overlapped, as shown in Figure 14c. In the second block, a rectangular window is used instead of the Raised Cosine to produce 128 output samples corresponding to the 416 input samples.

The phase estimator is the same as that shown in Figure 7, except that there are only 128 input samples, and the same description applies.

For both the LAN and cellular networks, the present system is designed to operate as a spread spectrum system preferably over such bands as are permitted, which at present are the 902 - 928 MHz band, 2.4 - 2.4835 GHz and 5.725 - 5.85 MHz. The carrier frequency in the local oscillator shown in Figures 5a, b and c will then be 915 MHz in the case of the 902 - 928 MHz band, and the frequencies used for modulation will be centered on this carrier frequency.

Alternative Embodiments

A person skilled in the art could make immaterial modifications to the invention described and claimed in this patent without departing from the essence of the invention.

For example, a system may consist of one or more central controllers (comparable to the Base Stations in the exemplary cellular system described) and some slave units (comparable to the portables).

The slave unit executes the commands it receives from the central controller. The commands may be requesting the slave unit to transmit a receive acknowledge, a status code or information that the slave has access to. The command may also be to relay the command or the information to another slave unit.

Vocoder Rate	vc slot structure		N=# of vc per frame	total # of FDvc =126 X N
18.77 Kbps	82 data sym.	3 control sym.	48	6,048
9.15 Kbps	40 data sym.	3 control sym.	95	11,970
6.18 Kbps	27 data sym.	3 control sym.	136	17,136

† vc = voice channel.
† FDvc = Full Duplex vc.

Table I

2064975

BS complexity of processor/ de-processor/estimator	vocoder rate	complexity of processor/ de-processor for portable
630 Mips	18.77 Kbps	1.35 Mips
	9.15 Kbps	1.29 Mips
	6.18 Kbps	1.26 Mips

† Mips = Mega Instruction Per Second.
† 1Instruction = 1complex add.+1complex multi.+1storage.

Table II

Vocoder rate	received SNR	BER
18.77 Kbps	27.8 dB	4×10^{-3}
9.15 Kbps	30.8 dB	2×10^{-3}
6.18 Kbps	31.33 dB	2×10^{-3}

Table III

100 MHz, 1 frame/cell				
# of cells	# of layers	urban area in Km^2	indoor area in Km^2	CIR in dB
126	7	24.6	0.355	37.9

(a)

50 MHz, 2 frames/cell				
# of cells	# of layers	urban area in Km^2	indoor area in Km^2	CIR in dB
63	5	11.94	0.172	25.8

(b)

5.86 MHz, 18 frames/cell				
# of cells	# of layers	urban area in Km^2	indoor area in Km^2	CIR in dB
7	2	1.43	0.02	13.8

(c)

Table IV

	100 MHz		50 MHz		5.86 MHz	
Vocoder Rate	N=# of FDvc/ frame	# of FDvc BS Tx/Rx (21X1XN)	N=# of FDvc/ frame	# of FDvc BS Tx/Rx (21X2XN)	N=# of FDvc/ frame	# of FDvc BS Tx/Rx (7X18XN)
18.77 Kbps	48	1008	24	1008	2	252
9.15 Kbps	95	1995	47	1974	5	630
6.18 Kbps	136	2856	68	2856	8	1008

Table V

2064975

Capacity = # of HDvc per 1.25 MHz per cell		
Analog FM	Our System (5.86 MHz, 6.18 Kbps vocoder)	
$\frac{1.25MHz}{30KHz} \times \frac{1}{7} \approx 6$	▪Dynamic Channel Allocation	2.6 x 144 x 2 x 1.25 / 6.25 = 150
	▪Voice Activation	150 X 1.5 = 225

† HDvc = Half Duplex vc

Table VI

2064975

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
CCCC
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
C           -1 TO +1 IN EACH DIRECTION
C
1         V1 = 2. * URAND(ISTRM) - 1.
          V2 = 2. * URAND(ISTRM) - 1.
C
C     SEE IF THEY ARE IN THE UNIT CIRCLE, IF THEY ARE NOT GO TO 1
C
          R = V1**2 + V2**2
          IF( R .GE. 1.) GO TO 1
C
C     NOW MAKE THE BOX-MULLER TRANSFORMATION TO GET TWO
NORMAL DEVIATES
C
          FAC = SQRT(-2. * ALOG(R)/R)
C
C     RETURN ONE AND SAVE ONE FOR NEXT TIME
C
          GSET = V1 * FAC
          BOXNRM = V2 * FAC
C
C     SET FLAG
C
          ISET = 1
C
      ELSE
C
C     WE HAVE AN EXTRA DEVIATE HANDY, SO RETURN IT
C
          BOXNRM = GSET
C
C     UNSET THE FLAG
C
          ISET = 0
C
      ENDIF
C
      RETURN
      END
```

APPENDIX A

Copyright @ 1991 Universeity Technologies Inc.

```
subroutine demod16(delta,cc)
complex cc
common /pie1/piover4,piover8,piover16

if(delta.ge.31.*piover16.or .delta.lt. 1.*piover16)cc=(+1,+1)
if(delta.ge. 1.*piover16.and.delta.lt. 3.*piover16)cc=(+1,-1)
if(delta.ge. 3.*piover16.and.delta.lt. 5.*piover16)cc=(-1,-1)
if(delta.ge. 5.*piover16.and.delta.lt. 7.*piover16)cc=(-1,+1)

if(delta.ge. 7.*piover16.and.delta.lt. 9.*piover16)cc=(-1,+3)
if(delta.ge. 9.*piover16.and.delta.lt.11.*piover16)cc=(-1,-3)
if(delta.ge.11.*piover16.and.delta.lt.13.*piover16)cc=(+1,-3)
if(delta.ge.13.*piover16.and.delta.lt.15.*piover16)cc=(+1,+3)

if(delta.ge.15.*piover16.and.delta.lt.17.*piover16)cc=(+3,+3)
if(delta.ge.17.*piover16.and.delta.lt.19.*piover16)cc=(+3,-3)
if(delta.ge.19.*piover16.and.delta.lt.21.*piover16)cc=(-3,-3)
if(delta.ge.21.*piover16.and.delta.lt.23.*piover16)cc=(-3,+3)

if(delta.ge.23.*piover16.and.delta.lt.25.*piover16)cc=(-3,+1)
if(delta.ge.25.*piover16.and.delta.lt.27.*piover16)cc=(+3,+1)
if(delta.ge.27.*piover16.and.delta.lt.29.*piover16)cc=(+3,-1)
if(delta.ge.29.*piover16.and.delta.lt.31.*piover16)cc=(-3,-1)

return
end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
subroutine demod4(delta,cc)
complex cc
common /pie1/piover4,piover8,piover16

if(delta.ge.0.*piover4.and.delta.lt.2.*piover4)cc=cmplx(+1,+1)
if(delta.ge.2.*piover4.and.delta.lt.4.*piover4)cc=cmplx(+1,-1)
if(delta.ge.4.*piover4.and.delta.lt.6.*piover4)cc=cmplx(-1,-1)
if(delta.ge.6.*piover4.and.delta.lt.8.*piover4)cc=cmplx(-1,+1)

return
end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
subroutine demod8(delta,cc)
complex cc
common /pie1/piover4,piover8,piover16

if(delta.ge.15.*piover8.or .delta.lt. 1.*piover8)cc=(+1,+1)
if(delta.ge. 1.*piover8.and.delta.lt. 3.*piover8)cc=(+1,-1)
If(delta.ge. 3.*piover8.and.delta.lt. 5.*piover8)cc=(-1,-1)
if(delta.ge. 5.*piover8.and.delta.lt. 7.*piover8)cc=(-1,+1)

if(delta.ge. 7.*piover8.and.delta.lt. 9.*piover8)cc=(-3,+1)
if(delta.ge. 9.*piover8.and.delta.lt.11.*piover8)cc=(+3,+1)
if(delta.ge.11.*piover8.and.delta.lt.13.*piover8)cc=(+3,-1)
if(delta.ge.13.*piover8.and.delta.lt.15.*piover8)cc=(-3,-1)

return
end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
c SUBROUTINE FFT() - REFER TO THE BOOK BY 'GONZALEZ & WINTZ'
c
c f is the input array to the fft

c f is also the output array from the fft with +ve freq. at the begining
c                                          and -ve freq. at the end.

c f1 is the output array from the fft with -ve freq. at the begining
c                                          and +ve freq. at the end.

c ind is the index array for f1.

c if ix=0 the F is the input array.
c if ix=1 the F1 is the input array.


      subroutine fft (ix,f,ind,f1, ln)
      integer ln, n, nv2, nm1, j, i, k, le, le1, ip,ind(2048)
      real pi
      complex f(2048), f1(2048),u, w, t
c
      pi = 4.0 * atan(1.0)
      n = 2**ln
      nv2 = n/2
      nm1 = n-1

      if(ix.eq.1)call deswap(f,f1,n)

      j = 1
      do 3 i= 1, nm1
       if (i.ge.j) go to 1
       t = f(j)
       f(j) = f(i)
       f(i) = t
1     k = nv2
2     if (k.ge.j) go to 3
      j = j-k
      k = k/2
      go to 2
3     j = j+k
      do 5 l= 1, ln
       le = 2**l
       le1 = le/2
       u = (1.0, 0.0)
       w = cmplx( cos(pi/le1), -sin(pi/le1) )
       do 5 j= 1, le1
        do 4 i= j, n, le
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
            ip = i + le1
            t = f(ip) * u
            f(ip) = f(i) - t
4           f(i) = f(i) + t
5       u = u * w
        do 6 i= 1, n
        f(i) = f(i) / float (n)
6       ind(i) = i-n/2-1

        call swap(f,f1,n)
        return
        end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
C*****************************************
C       Inverse fft transform
C*****************************************
c
c if ix=0, f is the input array to the Ifft with +ve freq. at the begining
c                                   and -ve freq. at the end.

c if ix=1, f1 is the input array to the Ifft with -ve freq. at the begining
c                                   and +ve freq. at the end.

      subroutine tdft (ix,xft,ind,xft1,nft)
      integer nft, i,  mft,ind(2048)
      complex xft(nft),xft1(2048), buff(2048), buff1(2048)
c
      mft = nint(alog10(float(nft))/alog10(2.0))
      print *, ' nft and mft are:', nft, mft

      if(ix.eq.1)call deswap(xft,xft1,nft)

      do 7 i= 1, nft
7       buff(i) = conjg(xft(i))

c       fft() does 1-D Fourier transform

      call fft(0,buff,ind,buff1, mft)

      do 8 i= 1, nft
        xft(i) = nft*conjg(buff(i))
8       xft1(i) = nft*conjg(buff1(i))

      return
      end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
C**********************************************************************

      subroutine swap(f,f1,n)
      complex f(2048), f1(2048)

      do i=-n/2,-1
       f1(i+n/2+1) = f(n+i+1)
      enddo

      do i=0,n/2-1
       f1(i+n/2+1) = f(i+1)
      enddo

      return
      end

C**********************************************************************

      subroutine deswap(f,f1,n)
      complex f(2048), f1(2048)

      do i=-n/2,-1
       f(n+i+1) = f1(i+n/2+1)
      enddo

      do i=0,n/2-1
       f(i+1) = f1(i+n/2+1)
      enddo

      return
      end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
      real pi,TWOPI,arRe(0:512),arIm(0:512),norm,p,
     &     curphase(0:512),oldph,ph,psi(0:512),RC(-2048:2048)
      integer ind(2048),counter
      complex z(0:512),r(0:512),c(0:512),d(0:512),cc
      complex F(2048),F1(2048)
      character string*50
      common /pie/pi
      common /pie1/piover4,piover8,piover16
      common /dec/no

      open(100,file='transmitted')
      open(200,file='received')
      open(400,file='BER')
      open(800,file='envelope')
      open(310,file='RC')
      open(350,file='data')
      open(351,file='random')
      pi = 4. * atan(1.0)
      piover4 = pi/4.
      piover8 = pi/8.
      piover16 = pi/16.
      TWOPI = 2. * pi
      read(*,102)string
      write(*,102)string
      read(*,*)number
      read(*,*)M
      read(*,*)MM
      read(*,*)no
102   format(a50)
C     open(101,file=string)

C     numb=255
      LL= MM/2 -1
      counter = 0
      call raisedcos(MM,RC)

C     do i=1,numb
c******* READING ******************************
C       read(101,*)p,arRe(i),arIm(i)
C       norm = ( 900. / p )**2
C       arRe(i) = arRe(i)/norm
C       arIm(i) = arIm(i)/norm
C       z(i)=cmplx(arRe(i),arIm(i))
C       write(800,*)i,sqrt(real(Z(i)*conjg(Z(i))))
C     enddo
C     z(0) =cmplx(arRe(1),arIm(1))
C     z(MM)=cmplx(arRe(numb),arIm(numb))
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
      do ii=1,number
C****** FIRST LOOP ******************************
      curphase(0) = urand(3)*TWOPI
      d(0) = cmplx(cos(curphase(0)),sin(curphase(0)))

C* STARTING THE DATA TRANSMISSION ***************
C****** DATA GENERATION ************************
      do 10 l=1,MM,1
        if(M.eq.4)call modu4(cc,delta)
        if(M.eq.8)call modu8(cc,delta)
        if(M.eq.16)call modu16(cc,delta)
        write(500,*)l,cc,delta

        c(l) = cc
        psi(l) = urand(4)*TWOPI
        write(351,*)psi(l)
        curphase(l) = curphase(l-1) + delta + psi(l)
        d(l) = cmplx(cos(curphase(l)),sin(curphase(l)))
      enddo

C****** MODULATOR ******************************
      do k=0,MM,1
        write(100,*)k,sqrt(real(d(k)*conjg(d(k))))
C       r(k) = d(k)*z(k)
        r(k) = d(k)
        write(451,*)k,r(k)
      enddo

C ****** FIND THE TIME RESPONSE ******************
      j=0
      do k=1,MM,1
        j=j+1
        F1(j) = r(k-1)
      enddo

     CALL tdft(1,F,ind,F1,MM)

      open(300,file='time_RCSF')
      open(301,file='time0')
      open(302,file='time_NOSF')

      call window(MM,LL,RC,F1,counter)

C     goto101

C****** DEMODULATOR ****************************
      oldph = atan2(aimag(r(0)),real(r(0)))
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
      do k=1,MM-1
        ph = atan2(aimag(r(k)),real(r(k)))
        delta = ph - oldph - psi(k)
        if(delta.gt.TWOPI)delta=delta-TWOPI
        if(delta.gt.TWOPI)delta=delta-TWOPI
        if(delta.gt.TWOPI)delta=delta-TWOPI
        if(delta.lt.0.)delta=delta+TWOPI
        if(delta.lt.0.)delta=delta+TWOPI
        if(delta.lt.0.)delta=delta+TWOPI

        if(M.eq.4)call demod4(delta,cc)
        if(M.eq.8)call demod8(delta,cc)
        if(M.eq.16)call demod16(delta,cc)
        write(501,*)k,cc,delta

        if(real(cc).ne.real(c(k)))errorRe=errorRe+1
        if(aimag(cc).ne.aimag(c(k)))errorIm=errorIm+1

        write(400,*)k,'  ',real(cc),real(c(k)),
     &              '  ',aimag(cc),aimag(c(k))
        oldph = ph
      enddo
      print*,'errorRe=',errorRe,'  ','errorIm=',errorIm
10    continue
101   continue
      enddo

      rewind(301)
      call quantize(LL,counter,prod)
      write(306,*)prod


      stop
      end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
subroutine modu16(cc,delta)
complex cc
common /pie1/piover4,piover8,piover16

s1 = urand(1)
s2 = urand(2)
s3 = urand(1)
s4 = urand(2)
if(s1.ge.0.5)s1=+1
if(s1.lt.0.5)s1=-1
if(s2.ge.0.5)s2=+1
if(s2.lt.0.5)s2=-1
if(s3.ge.0.5)s3=+1
if(s3.lt.0.5)s3=-1
if(s4.ge.0.5)s4=+1
if(s4.lt.0.5)s4=-1
cc = cmplx(s1+2*s3,s2+2*s4)
write(350,*)cc

if(cc.eq.(+1,+1))delta=0.*piover8
if(cc.eq.(+1,-1))delta=1.*piover8
if(cc.eq.(-1,-1))delta=2.*piover8
if(cc.eq.(-1,+1))delta=3.*piover8

if(cc.eq.(-1,+3))delta=4.*piover8
if(cc.eq.(-1,-3))delta=5.*piover8
if(cc.eq.(+1,-3))delta=6.*piover8
if(cc.eq.(+1,+3))delta=7.*piover8

if(cc.eq.(+3,+3))delta=8.*piover8
if(cc.eq.(+3,-3))delta=9.*piover8
if(cc.eq.(-3,-3))delta=10.*piover8
if(cc.eq.(-3,+3))delta=11.*piover8

if(cc.eq.(-3,+1))delta=12.*piover8
if(cc.eq.(+3,+1))delta=13.*piover8
if(cc.eq.(+3,-1))delta=14.*piover8
if(cc.eq.(-3,-1))delta=15.*piover8

return
end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
subroutine modu4(cc,delta)
complex cc
common /pie1/piover4,piover8,piover16

s1 = urand(1)
s2 = urand(2)
if(s1.ge.0.5)s1=+1
if(s1.lt.0.5)s1=-1
if(s2.ge.0.5)s2=+1
if(s2.lt.0.5)s2=-1
cc = cmplx(s1,s2)
write(350,*)cc

if(cc.eq.(+1,+1))delta=1.*piover4
if(cc.eq.(+1,-1))delta=3.*piover4
if(cc.eq.(-1,-1))delta=5.*piover4
if(cc.eq.(-1,+1))delta=7.*piover4

return
end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
subroutine modu8(cc,delta)
complex cc
common /pie1/piover4,piover8,piover16

s1 = urand(1)
s2 = urand(2)
s3 = urand(1)
if(s1.ge.0.5)s1=+1
if(s1.lt.0.5)s1=-1
if(s2.ge.0.5)s2=+1
if(s2.lt.0.5)s2=-1
if(s3.ge.0.5)s3=+1
if(s3.lt.0.5)s3=-1
cc = cmplx(s1+2*s3,s2)
pp = urand(4)*TWOPI
write(350,*)cc

if(cc.eq.(+1,+1))delta=0.*piover4
if(cc.eq.(+1,-1))delta=1.*piover4
if(cc.eq.(-1,-1))delta=2.*piover4
if(cc.eq.(-1,+1))delta=3.*piover4

if(cc.eq.(-3,+1))delta=4.*piover4
if(cc.eq.(+3,+1))delta=5.*piover4
if(cc.eq.(+3,-1))delta=6.*piover4
if(cc.eq.(-3,-1))delta=7.*piover4

return
end
```

APPENDIX A

Copyright © 1991 University Technologies Inc.

```
      subroutine quantize(LL,counter,prod)
      integer Fre,Fim,counter

      xmax=-100.
      xmin=100.
      do j=1,counter
        read(301,*)i,arRe,arIm
        if(arRe.gt.xmax)xmax=arRe
        if(arIm.gt.xmax)xmax=arIm
        if(arRe.lt.xmin)xmin=arRe
        if(arIm.lt.xmin)xmin=arIm
      enddo
      if(abs(xmax).gt.abs(xmin))xxx=abs(xmax)
      if(abs(xmax).lt.abs(xmin))xxx=abs(xmin)
      prod = 128./xxx

      sumRe = 0.
      sumIm = 0.
      surRe = 0.
      surIm = 0.
      rewind(301)
      do j=1,counter
        read(301,*)i,arRe,arIm
        Fre = 128 + prod*arRe
        Fim = 128 + prod*arIm
C       if(Fre.gt.255)Fre=255
C       if(Fre.lt.0)Fre=0
C       if(Fim.gt.255)Fim=255
C       if(Fim.lt.0)Fim=0
        write(401,*)i,Fre,Fim
C       write(402,*)j,Fre
C       write(403,*)j,Fim
        sumRe = sumRe + (arRe-(Fre-128.)/prod)**2
        sumIm = sumIm + (arIm-(Fim-128.)/prod)**2
        surRe = surRe + (arRe)**2
        surIm = surIm + (arIm)**2
      enddo
      print*,'quantization noise=',(sumRe + sumIm)/(surRe + surIm)
      print*,prod

      return
      end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
subroutine raisedcos(MM,RC)
real RC(-2048:2048),pi
common /pie/pi

do i=-MM/2-MM/8,-MM/2+MM/8
  t = -float(i)/float(MM)
  RC(i)=(0.5*(1.-sin(4.*pi*(t-0.5))))
  write(310,*)i,RC(i)
enddo

do i=-MM/2+MM/8+1,MM/2-MM/8-1
  t = float(i)/float(MM)
  RC(i)=1.
  write(310,*)i,RC(i)
enddo

do i=MM/2-MM/8,MM/2+MM/8
  t = float(i)/float(MM)
  RC(i)=(0.5*(1.-sin(4.*pi*(t-0.5))))
  write(310,*)i,RC(i)
enddo

return
end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
subroutine demod4(delta,wRe,wIm)

y = cmplx(cos(delta),sin(delta))
if(real(y).ge.0)then
  wRe=+1.
else
  wRe=-1.
endif

if(aimag(y).ge.0)then
  wIm=+1.
else
  wIm=-1.
endif

return
end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    NAME           : URAND(ISTRM) - FUNCTION          +
C                                                      +
C    FUNCTION       : GENERATES STANDARD UNIFORMLY DISTRIBUTED
+
C                     RANDOM NUMBERS .                 +
C                     ( USES THE RECURSION:  SEED(ISTRM) =      +
C                          16807 * SEED(ISTRM) MOD (2**(31) - 1) ) +
C                     SOME COMPILERS REQUIRE THE DECLARATION:
+
C                          INTEGER*4 ISTRM, K1          +
C                     REF :                            +
C                                                      +
C    CALLING PROG / SUBPROG :                          +
C                                                      +
C    CALLED SUBPROG      : NIL                         +
C                                                      +
C    INPUTS         : ISTRM - THE STREAM NUMBER        +
C                          0 < SEED(ISTRM) < 2147483647      +
C                                                      +
C    OUTPUTS        : STD. UNIFORMLY DISTRIBUTED RANDOM NUMBERS
+
C                          NEW VALUE OF SEED(ISTRM)    +
C                                                      +
C    PRINCIPAL VARIABLES  :                            +
C                                                      +
C    DATE           : 26 APRIL , 1985.                 +
C                                                      +
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      FUNCTION URAND(ISTRM)
C
C     DECLARATION :
C
        INTEGER SIZE3
      PARAMETER(SIZE3 = 20)
      REAL URAND
      INTEGER SEED(SIZE3),ISTRM,ISTRG,K1
C
        save seed
        data seed(1)/456789/
        data seed(2)/1037625857/
        data seed(3)/102899405/
        data seed(4)/535417/
        data seed(5)/211229747/
        data seed(6)/2952853/
        data seed(7)/2987570/
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
      data seed(8)/45376256/
      data seed(9)/17890857/
C
C     GENERATE A U(0,1) VARIATE
C
      ISTRG = ISTRM
      K1 = SEED(ISTRG)/127773
      SEED(ISTRG) = 16807 * ( SEED(ISTRG) - K1 * 127773) - K1 * 2836
      IF( SEED(ISTRG) .LT. 0) SEED(ISTRG) = SEED(ISTRG) + 2147483647
      URAND = FLOAT(SEED(ISTRG)) * 4.656612875E-10
C
      RETURN
      END
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
      subroutine window(MM,LL,RC,F,counter)
      complex F(2048),F0(-2048:2048),Fm(-2048:2048),Fp(-2048:2048),
     &     Fsum(-2048:2048),FRC(-2048:2048)
      real RC(-2048:2048)
      integer counter,count1
      common /dec/no
      save count1
      data count1/0/

      count1 = count1 + 1
      do i=-MM-LL,MM+LL+1
        F0(i) = (0.,0.)
        Fm(i) = (0.,0.)
        Fp(i) = (0.,0.)
      enddo

      do i=-LL,LL+1
        F0(i)    = F(i+LL+1)
        Fm(i-MM) = F0(i)
        Fp(i+MM) = F0(i)
        write(304,*)i,F0(i)
      enddo

      do i=-MM-LL,MM+LL+1
        Fsum(i) = F0(i) + Fm(i) + Fp(i)
        write(302,*)i,sqrt((real(Fsum(i)))**2+(aimag(Fsum(i)))**2)
      enddo

      do i=-LL - (LL+1)/4-8,-LL - (LL+1)/4-1
        counter = counter + 1
        FRC(i) = cmplx(0.,0.)
        write(301,*)counter,0.,0.
      enddo
      do i=-LL - (LL+1)/4,LL+1 + (LL+1)/4
        counter = counter + 1
        FRC(i) = Fsum(i)*RC(i)
        write(300,*)counter,sqrt((real(FRC(i)))**2+(aimag(FRC(i)))**2)
        write(301,*)counter,real(FRC(i)),aimag(FRC(i))
        write(202,*)counter,real(FRC(i))
        write(203,*)counter,aimag(FRC(i))
      enddo
      do i=LL+1 + (LL+1)/4+1,LL+1 + (LL+1)/4+8
        counter = counter + 1
        FRC(i) = cmplx(0.,0.)
        write(301,*)counter,0.,0.
      enddo
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
if(count1.eq.no)then
do i=-LL - (LL+1)/4-8,LL+1 + (LL+1)/4+8
  counter = counter + 1
  FRC(i) = cmplx(0.,0.)
  write(301,*)counter,0.,0.
  write(202,*)counter,real(FRC(i))
  write(203,*)counter,aimag(FRC(i))
enddo
endif

return
end
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
FFLAGS =
FFILES = main.f urand.f fft.f window.f raisedcos.f quantize.f \
modu4.f moud8.f modu16.f demod4.f demod8.f demod16.f

OFILES = main.o urand.o fft.o window.o raisedcos.o quantize.o \
modu4.o modu8.o modu16.o demod4.o demod8.o demod16.o

a.out: ${OFILES}
	f77 ${FFLAGS} ${OFILES}
```

APPENDIX A

Copyright @ 1991 University Technologies Inc.

```
../../../../../HF_LF/TWOCHANNEL/900/test.1
9
16
64
9
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
CCCC
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
subroutine demod4(delta,cc)
complex cc
common /pie1/piover4,piover8,piover16

if(delta.ge.0.*piover4.and.delta.lt.2.*piover4)cc=cmplx(+1,+1)
if(delta.ge.2.*piover4.and.delta.lt.4.*piover4)cc=cmplx(+1,-1)
if(delta.ge.4.*piover4.and.delfa.lt.6.*piover4)cc=cmplx(-1,-1)
if(delta.ge.6.*piover4.and.delta.lt.8.*piover4)cc=cmplx(-1,+1)

return
end
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
subroutine demod8(delta,cc)
complex cc
common /ple1/piover4,piover8,piover16

if(delta.ge.15.*piover8.or .delta.lt. 1.*piover8)cc=(+1,+1)
if(delta.ge. 1.*piover8.and.delta.lt. 3.*piover8)cc=(+1,-1)
if(delta.ge. 3.*piover8.and.delta.lt. 5.*piover8)cc=(-1,-1)
if(delta.ge. 5.*piover8.and.delta.lt. 7.*piover8)cc=(-1,+1)

if(delta.ge. 7.*piover8.and.delta.lt. 9.*piover8)cc=(-3,+1)
if(delta.ge. 9.*piover8.and.delta.lt.11.*piover8)cc=(+3,+1)
if(delta.ge.11.*piover8.and.delta.lt.13.*piover8)cc=(+3,-1)
if(delta.ge.13.*piover8.and.delta.lt.15.*piover8)cc=(-3,-1)

return
end
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
      subroutine dewindow(MM,numb,F)
      integer shift
      complex F(2048),F0(-2048:2048),Fm(-2048:2048),
     &    Fsum(-2048:2048),Fp(-2048:2048),fprime
      common /SHHH/shift
      common /SDHH/prod

      do i=1,numb
        read(101,*)j,arRe,arIm
        F(i) = cmplx(arRe-shift,arIm-shift)
      enddo

      if(coun1.eq.no)then
      do i=1,numb
        read(101,*)j,arRe,arIm
      enddo
      endif

      LL=MM/2-1
      do i=-MM-numb/2+1,MM+numb/2
        F0(i) = (0.,0.)
        Fm(i) = (0.,0.)
        Fp(i) = (0.,0.)
      enddo

      do i=-numb/2+1,numb/2
        F0(i)   = F(i+numb/2)
        Fm(i-MM) = F0(i)
        Fp(i+MM) = F0(i)
      enddo

      do i=-MM-numb/2+1,MM+numb/2
        Fsum(i) = F0(i) + Fm(i) + Fp(i)
      enddo

      sum = 0.
      sup = 0.
      do i=-LL,LL+1
        write(303,*)i,sqrt((real(Fsum(i)))**2
     &          +(aimag(Fsum(i)))**2)
        F(i+LL+1) = Fsum(i)
        write(304,*)i,Fsum(i)

C       read(604,*)j,Fprime
C       sum = sum + conjg(Fsum(i)/prod-Fprime)*
C    &          (Fsum(i)/prod-Fprime)
C       write(602,*)i,abs(real(Fsum(i)/prod))
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
C   &            -abs(real(Fprime))
C        write(603,*)i,abs(aimag(Fsum(i)/prod))
C   &            -abs(aimag(Fprime))
C        write(605,*)i,Fsum(i)/prod-Fprime
C
C        sup = sup + conjg(Fprime)*(Fprime)
         enddo
C        print*,'quantization error=',sum,sup,sum/sup

         return
         end
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
c SUBROUTINE FFT() - REFER TO THE BOOK BY 'GONZALEZ & WINTZ'
c
c f is the input array to the fft

c f is also the output array from the fft with +ve freq. at the begining
c                                   and -ve freq. at the end.

c f1 is the output array from the fft with -ve freq. at the begining
c                                   and +ve freq. at the end.

c ind is the index array for f1.

c if ix=0 the F is the input array.
c if ix=1 the F1 is the input array.


      subroutine fft (ix,f,ind,f1, ln)
      integer ln, n, nv2, nm1, j, i, k, le, le1, ip,ind(2048)
      real pi
      complex f(2048), f1(2048),u, w, t
c
      pi = 4.0 * atan(1.0)
      n = 2**ln
      nv2 = n/2
      nm1 = n-1

      if(ix.eq.1)call deswap(f,f1,n)

      j = 1
      do 3 i= 1, nm1
       if (i.ge.j) go to 1
       t = f(j)
       f(j) = f(i)
       f(i) = t
1     k = nv2
2     if (k.ge.j) go to 3
       j = j-k
       k = k/2
       go to 2
3     j = j+k
      do 5 l= 1, ln
       le = 2**l
       le1 = le/2
       u = (1.0, 0.0)
       w = cmplx( cos(pi/le1), -sin(pi/le1) )
       do 5 j= 1, le1
        do 4 i= j, n, le
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
            ip = i + le1
            t = f(ip) * u
            f(ip) = f(i) - t
4           f(i) = f(i) + t
5       u = u * w
        do 6 i= 1, n
        f(i) = f(i) / float (n)
6       ind(i) = i-n/2-1

        call swap(f,f1,n)
        return
        end
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
C*********************************************
C      Inverse fft transform
C*********************************************
c
c if ix=0, f is the input array to the ifft with +ve freq. at the begining
c                                    and -ve freq. at the end.

c if ix=1, f1 is the input array to the ifft with -ve freq. at the begining
c                                    and +ve freq. at the end.

      subroutine tdft (ix,xft,ind,xft1,nft)
      integer nft, i,  mft,ind(2048)
      complex xft(nft),xft1(2048), buff(2048), buff1(2048)
c
      mft = nint(alog10(float(nft))/alog10(2.0))
      print *, ' nft and mft are:', nft, mft

      if(ix.eq.1)call deswap(xft,xft1,nft)

      do 7 i= 1, nft
7       buff(i) = conjg(xft(i))

c       fft() does 1-D Fourier transform

      call fft(0,buff,ind,buff1, mft)

      do 8 i= 1, nft
        xft(i) = nft*conjg(buff(i))
8       xft1(i) = nft*conjg(buff1(i))

      return
      end
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
C**********************************************************************

      subroutine swap(f,f1,n)
      complex f(2048), f1(2048)

      do i=-n/2,-1
       f1(i+n/2+1) = f(n+i+1)
      enddo

      do i=0,n/2-1
       f1(i+n/2+1) = f(i+1)
      enddo

      return
      end
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
C**********************************************************************

      subroutine deswap(f,f1,n)
      complex f(2048), f1(2048)

      do i=-n/2,-1
       f(n+i+1) = f1(i+n/2+1)
      enddo

      do i=0,n/2-1
       f(i+1) = f1(i+n/2+1)
      enddo

      return
      end
```

APPENDIX B

Copyright @ 1991 Universeity Technologies Inc.


```
      real pi,TWOPI,oldph,ph,psi(0:512)
      complex r(0:512),c(0:512),cc,cl
      complex F(2048),F1(2048)
      integer ind(2048),shift
      character string*40,string1*40,string2*40
      common /pie/pi
      common /pie1/piover4,piover8,piover16
      common /SHHH/shift
      common /SDHH/prod

      open(100,file='transmitted')
      open(200,file='received')
      open(400,file='BER')
      open(800,file='envelope')
      open(310,file='RC')
      pi = 4. * atan(1.0)
      piover4 = pi/4.
      piover8 = pi/8.
      piover16 = pi/16.
      TWOPI = 2. * pi
      read(*,102)string
      write(*,102)string
      read(*,102)string1
      write(*,102)string1
      read(*,102)string2
      write(*,102)string2
      read(*,*)number
      read(*,*)M
      read(*,*)MM
      read(*,*)ni
102   format(a40)
      open(101,file=string)
      open(350,file=string1)
      open(351,file=string2)
      open(604,file='../fort.304')
      open(704,file='../fort.501')
      shift = 0
      prod = 1.
      if(string.eq.'../fort.401')shift=128
      if(string.eq.'../fort.401')read(306,*)prod


      numb=MM*1.25 + 16

c******* READING ******************************

      sum = 0.
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
      do ii=1, number
      call dewindow(MM,numb,F)

      call fft(0,F,ind,F1,ni)

      j=0
      do k=0,MM-1,1
        j=j+1
        r(k) = F1(j)
        write(451,*)k,r(k)
        read(350,*)c(k+1)
        read(351,*)psi(k+1)
      enddo

C****** DEMODULATOR ***************************
      oldph = atan2(aimag(r(0)),real(r(0)))
      do k=1,MM-1
        ph = atan2(aimag(r(k)),real(r(k)))
        delta = ph - oldph - psi(k) + pi
        if(delta.gt.TWOPI)delta=delta-TWOPI
        if(delta.gt.TWOPI)delta=delta-TWOPI
        if(delta.gt.TWOPI)delta=delta-TWOPI
        if(delta.lt.0.)delta=delta+TWOPI
        if(delta.lt.0.)delta=delta+TWOPI
        if(delta.lt.0.)delta=delta+TWOPI

        if(M.eq.4)call demod4(delta,cc)
        if(M.eq.8)call demod8(delta,cc)
        if(M.eq.16)call demod16(delta,cc)
        write(501,*)k,cc,delta
        read(704,*)i,cl,deltap
        diff = delta-deltap
        if(diff.gt.+pi)diff=diff-TWOPI
        if(diff.lt.-pi)diff=diff+TWOPI
        if(k.ne.MM/2.and.k.ne.MM/2+1)sum = sum + diff**2
        write(606,*)k,diff

        if(real(cc).ne.real(c(k)))errorRe=errorRe+1
        if(aimag(cc).ne.aimag(c(k)))errorIm=errorIm+1

        write(400,*)k,'  ',real(cc),real(c(k)),
     &              '  ',aimag(cc),aimag(c(k))
        oldph = ph
      enddo
      print*,'errorRe=',errorRe,'  ','errorIm=',errorIm
      print*,'sum=',sum/float(MM-3)
10    continue
      enddo
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
stop
end
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
FFLAGS =
FFILES = main.f fft.f dewindow.f demod4.f demod8.f demod16.f

OFILES = main.o fft.o dewindow.o demod4.o demod8.o demod16.o

a.out: ${OFILES}
	f77 ${FFLAGS} ${OFILES}
```

APPENDIX B

Copyright @ 1991 University Technologies Inc.


```
../fort.401
../data
../random
9
16
64
6
```

APPENDIX C

Copyright @ 1991 AGT Limited

SOFTWARE FOR SIMULATION ON DSP32C

The following software program was developed and run using D3SIM of AT&T. It simulates the operation of the equalizer on a DSP32C.

```
#include <math.h>
#include <stdio.h>
#include <libap.h>

/*****************************************************************************

                              main program

*****************************************************************************/
double twopi=6.283185372;
/*  List the original transmitted symbols  */
int jj2[514] =(   -1  ,  -1,
    1  ,   1,
    1  ,  -1,
    1  ,  -1,
    1  ,  -1,
   -1  ,  -1,
   -1  ,   1,
    1  ,   1,
   -1  ,   1,
    1  ,  -1,
   -1  ,  -1,
    1  ,   1,
   -1  ,  -1,
   -1  ,  -1,
   -1  ,   1,
   -1  ,  -1,
   -1  ,   1,
   -1  ,  -1,
    1  ,   1,
   -1  ,  -1,
    1  ,  -1,
   -1  ,  -1,
    1  ,  -1,
    1  ,  -1,
   -1  ,   1,
   -1  ,   1,
    1  ,  -1,
    1  ,  -1,
   -1  ,  -1,
    1  ,  -1,
   -1  ,  -1,
   -1  ,  -1,
   -1  ,   1,
```

APPENDIX C

Copyright @ 1991 AGT Limited

```
-1 , -1,
-1 ,  1,
-1 , -1,
 1 , -1,
-1 ,  1,
-1 ,  1,
-1 ,  1,
 1 , -1,
 1 ,  1,
-1 , -1,
-1 ,  1,
 1 , -1,
 1 ,  1,
 1 ,  1,
-1 ,  1,
 1 , -1,
-1 , -1,
 1 , -1,
-1 ,  1,
-1 , -1,
 1 ,  1,
 1 , -1,
-1 ,  1,
 1 ,  1,
-1 , -1,
-1 , -1,
-1 ,  1,
 1 , -1,
 1 , -1,
-1 ,  1,
 1 , -1,
-1 , -1,
-1 ,  1,
-1 , -1,
 1 , -1,
-1 , -1,
 1 ,  1,
-1 , -1,
 1 , -1,
 1 , -1,
-1 , -1,
-1 , -1,
-1 , -1,
-1 ,  1,
 1 , -1,
-1 , -1,
 1 ,  1,
-1 , -1,
-1 , -1,
-1 ,  1,
 1 ,  1,
 1 , -1,
-1 , -1,
```

APPENDIX C

Copyright @ 1991 AGT Limited

```
-1 , -1,
 1 , -1,
 1 , -1,
 1 ,  1,
-1 , -1,
 1 , -1,
-1 ,  1,
-1 , -1,
 1 , -1,
-1 , -1,
 1 , -1,
 1 ,  1,
 1 , -1,
-1 ,  1,
-1 ,  1,
 1 ,  1,
-1 , -1,
-1 , -1,
-1 , -1,
 1 , -1,
 1 ,  1,
 1 ,  1,
 1 ,  1,
-1 ,  1,
 1 ,  1,
 1 ,  1,
 1 ,  1,
-1 , -1,
-1 , -1,
 1 , -1,
 1 ,  1,
 1 , -1,
-1 , -1,
 1 , -1,
 1 ,  1,
 1 , -1,
 1 , -1,
-1 , -1,
-1 ,  1,
 1 ,  1,
-1 ,  1,
 1 , -1,
-1 ,  1,
 1 ,  1,
-1 , -1,
 1 ,  1,
 1 ,  1,
-1 , -1,
-1 ,  1,
 1 , -1,
-1 , -1,
 1 ,  1,
-1 , -1,
```

APPENDIX C

Copyright @ 1991 AGT Limited

```
 1 ,  1,
 1 , -1,
 1 , -1,
 1 , -1,
-1 , -1,
 1 ,  1,
 1 , -1,
-1 ,  1,
 1 ,  1,
 1 , -1,
 1 , -1,
-1 ,  1,
-1 ,  1,
 1 , -1,
-1 ,  1,
 1 , -1,
-1 , -1,
-1 , -1,
-1 ,  1,
-1 ,  1,
-1 , -1,
-1 , -1,
-1 ,  1,
-1 ,  1,
-1 ,  1,
-1 ,  1,
-1 , -1,
-1 , -1,
 1 ,  1,
 1 , -1,
 1 , -1,
 1 ,  1,
-1 ,  1,
-1 ,  1,
-1 , -1,
 1 ,  1,
-1 , -1,
 1 , -1,
 1 ,  1,
 1 ,  1,
-1 ,  1,
 1 ,  1,
-1 ,  1,
-1 , -1,
-1 , -1,
-1 , -1,
 1 , -1,
-1 ,  1,
 1 ,  1,
-1 , -1,
 1 ,  1,
 1 , -1,
-1 ,  1,
```

APPENDIX C

Copyright @ 1991 AGT Limited

```
 1 , -1,
-1 , -1,
 1 ,  1,
 1 , -1,
-1 , -1,
 1 ,  1,
-1 ,  1,
-1 ,  1,
 1 , -1,
 1 ,  1,
 1 ,  1,
 1 , -1,
-1 , -1,
 1 , -1,
-1 , -1,
 1 ,  1,
 1 , -1,
-1 , -1,
-1 , -1,
-1 ,  1,
-1 ,  1,
 1 ,  1,
 1 ,  1,
-1 ,  1,
 1 , -1,
-1 , -1,
-1 ,  1,
 1 , -1,
 1 , -1,
 1 , -1,
 1 , -1,
 1 ,  1,
-1 ,  1,
-1 ,  1,
-1 , -1,
-1 ,  1,
-1 , -1,
 1 , -1,
 1 , -1,
 1 , -1,
 1 , -1,
 1 , -1,
 1 ,  1,
-1 , -1,
-1 ,  1,
-1 ,  1,
 1 ,  1,
-1 ,  1,
-1 , -1,
 1 , -1,
 1 , -1,
 1 ,  1,
 1 , -1,
```

APPENDIX C

Copyright @ 1991 AGT Limited

```
 -1 , -1,
  1 ,  1,
 -1 , -1,
  1 ,  1,
  1 , -1,
  1 ,  1,
  1 , -1,
 -1 , -1,
  1 , -1,
 -1 , -1,
 -1 ,  1,
  1 ,  1
  };
/*  List the received I and Q */
double z[514]={
  1.8826647996902  ,0.57904273271561,
 -0.98167502880096 , -1.7151488065720,
 -1.9221851825714 , -0.45452463626862,
 -1.6370518207550  , 1.08983910083 77,
 -0.31615784764290  , 1.9248703718185,
 -1.5235337018967  , 1.1806204319000,
  0.16620002686977  , -1.8899723291397,
 -1.3761224746704  , 1.2521643638611,
  7.1681058034301E-03  , -1.8177816867828,
 -1.3031834363937  , -1.1972546577454,
 -0.15820230543613  , -1.7096272706985,
 -0.99022841453552  , 1.3329787254333,
 -1.5678428411484 , 0.32700437307358,
 -1.3414373397827 , -0.75913363695145,
  1.3957004547119 , -0.49621558189392,
  1.3290159702301 , 0.50875544548035,
 -1.1973139047623 , 0.66275292634964,
 -1.2967063188553 , -0.24445547163486,
  0.82353335618973 , 0.97751879692078,
  2.7966260910034E-02  , 1.2459809780121,
  0.97553628683090 , 0.74172759056091,
  0.30242365598679  , 1.1787307262421,
  1.1158645153046 , 0.49576562643051,
  1.0971860885620 , -0.57270908355713,
 -0.24569725990295  , 1.2418062686920,
 -0.83266228437424  , -1.0038361549377,
 -1.3508954048157  , 2.3582908324897E-03,
 -0.90133684873581  , 1.0763428211212,
 -1.4409677982330 , 0.24240809679031,
 -0.79242151975632  , 1.2981982231140,
 -1.5102118253708 , 0.46876969933510,
 -1.4932200908661 , -0.67981052398682,
  1.5572143793106 , -0.67623662948608,
  1.6570891141891 , 0.56612873077393,
 -1.5809999704361 , 0.86023199558258,
 -1.7863020896912 , -0.45382413268089,
 -1.5810590982437  , 1.0169448852539,
 0.34510406851768  , -1.8782730102539,
```

APPENDIX C

Copyright @ 1991 AGT Limited

1.1434341669083 , 1.5573689937592,
-1.9314140081406 , -0.24187317490578,
-1.5104018449783 , 1.2377147674561,
1.9451805353165 , 0.14569547772408,
1.2988004684448 , 1.4411246776581,
-1.9200929403305 , -5.77612631022938-02,
-1.3509848117828 , 1.3267296552658,
1.8577233552933 , -2.11265552788978-02,
-1.3225439786911 , -1.2418869733810,
1.7606525421143 , -9.05509814620028-02,
1.1161555051804 , -1.2882466316223,
1.6323974132538 , -0.15046791732311,
0.97648859024048 , -1.2267235517502,
0.20117856562138 , 1.4773093461990,
-0.82589977979660 , 1.1416391134262,
1.3004480600357 , -0.24329216778278,
0.66765332221985 , -1.0373107194901,
0.27767503261566 , 1.1074353456497,
0.50518971681595 , -0.91855829954147,
0.90428864955902 , -0.30539751052856,
0.79054325819016 , 0.34204551577568,
-0.69724625349045 , 0.32767096161842,
-0.18177013099194 , 0.65859514474869,
0.34578496217728 , 0.49258133769035,
-0.52803617715836 , -2.78416797518738-02,
-0.29641786217690 , 0.36104345321655,
-0.40400594472885 , 0.11642142385244,
0.11455089598894 , -0.37470200657845,
0.29129487276077 , -0.24794004857540,
-4.77271266281608-02 , -0.38791200518608,
0.19418933987617 , -0.36395436525345,
-0.40166890621185 , 0.18578046560287,
-0.46209335327148 , -0.11633513122797,
-0.41677090525627 , 0.29576727747917,
-6.06224350631248-02 , 0.540437161922245,
-0.43378528952599 , 0.37475159764290,
-0.59756731987000 , -2.90964897722018-02,
-0.42078813910484 , -0.45302680134773,
0.63260388374329 , 2.28966325521478-02,
0.47454643249512 , -0.43297693133354,
0.64523047208786 , 4.22249808907518-02,
-0.41148558259010 , -0.49813264608383,
8.63467678427708-02 , -0.63570129871368,
0.52332174777985 , -0.35753899812698,
-0.15361997485161 , 0.60483527183533,
0.54942035675049 , -0.27337777614594,
0.24155624210835 , -0.55399334430695,
0.57553601264954 , -0.16218262910843,
0.48504012823105 , 0.34690761566162,
0.60061538219452 , -2.79741454869518-02,
0.46577885746956 , -0.40029266476631,
-0.62349128723145 , -0.12451778352261,
-0.30245357751846 , -0.59375917911530,

APPENDIX C

Copyright @ 1991 AGT Limited

-0.64292871952057 , -0.29000273346901,
0.72607111930847 , -0.19453555345535,
0.65767973661423 , 0.46279823780060,
0.85773074626923 , -7.97747746109961E-02,
0.66653263568878 , 0.63700795173645,
0.98371291160583 , 3.84636558592321E-02,
-0.66836547851562 , -0.80670833587646,
-1.09911823272711 , -0.15678122639656,
0.96613508462906 , -0.66218781471252,
-0.27184593677521 , 1.19933533668521,
1.1098629236221 , -0.64718896150589,
1.2801918983459 , 0.38049080967903,
0.62276816368103 , 1.2329781055450,
-0.47980448603630 , 1.3380939960480,
0.58856463432312 , 1.3312312364578,
0.56721973419189 , -1.3701412677765,
-1.4011698961258 , 0.54447442293167,
1.3742272853851 , 0.64058619737625,
-1.4402492046356 , 0.49066120386124,
1.3491046428680 , 0.69823592901230,
-0.42755731940269 , -1.4469082355499,
-0.73903095722198 , 1.2944310903549,
-1.4206207990646 , 0.35585281252861,
-1.2107812166214 , -0.76239341497421,
-1.3619115352631 , 0.27647992968559,
1.0996322631836 , 0.76832592487335,
1.2723374366760 , -0.19058661162853,
0.96331912279129 , 0.75740247964859,
1.1544399261475 , -9.95059981942181E-02,
-0.80496501922607 , -0.73074960708618,
-1.0116646289825 , 4.71803499385711E-03,
-0.69000792503357 , 0.62838578224182,
-0.84825158119202 , -9.21911448240281E-02,
0.63727533817291 , -0.43797433376312,
-0.66910350322723 , -0.18957243859768,
0.57503563165665 , -0.23856720328331,
0.28575962781906 , -0.47963011264801,
3.52981239557271E-02 , 0.50607603788376,
0.37911093235016 , -0.28557923436165,
0.43339198827744 , 0.16655600070953,
-9.28561240434651E-02 , -0.46805140376091,
-0.36173328757286 , 0.36008602380753,
-0.55111116170883 , -9.26620736718181E-02,
0.54514288902283 , -0.28926151990891,
0.26530939340591 , -0.62695705890656,
0.71201461553574 , -0.22391571104527,
-0.69442093372345 , -0.41981008648872,
-0.16683530807495 , -0.85803741216660,
-0.55144131183624 , 0.75251996517181,
0.12049946933985 , 0.97948336601257,
0.80047225952148 , 0.65618216991425,
1.0733104944229 , -8.69883149862291E-02,
0.83770149946213 , 0.73083782196045,

APPENDIX C

Copyright @ 1991 AGT Limited

-6.79078176617 62E-02 , -1.1372458934784,
-0.86384069919586 , -0.77313911914825,
1.1698449850082 , -6.4328491687775E-02,
-0.87872487306595 , -0.78181076049805,
-1.1705212593079 , 7.6740458607674E-02,
-0.75661289691925 , 0.88237941265106,
-0.10502639412880 , -1.1395549774170,
0.87500441074371 , 0.69834393262863,
1.0780709981918 , -0.14845751225948,
-0.60881644487381 , 0.85695564746857,
0.20570844411850 , 0.98799550533295,
-0.49079838395119 , 0.82871961593628,
-0.87198609113693 , 0.27489036321640,
0.34792473912239 , -0.79089194536209,
0.35360851883888 , 0.73334228992462,
-0.18458327651024 , 0.74415004253387,
-0.57590019702911 , 0.43903124332428,
5.7776970788836E-03 , -0.68923103809357,
0.52798008918762 , 0.40390974283218,
-0.62690776586533 , 0.18303082883358,
0.22190481424332 , -0.61702972650528,
0.55797028541565 , -0.37609174847603,
0.70261883735657 , 3.4565746784210E-02,
-0.56754517555237 , -0.48320794105530,
-0.78116804361343 , 0.15341848134995,
-0.40339776873589 , 0.75160259008408,
0.84919923543930 , -0.33748552203178,
-0.31929352879524 , 0.92272567749023,
-0.51332873106003 , -0.90345317125320,
0.23162299394608 , -1.0757941007614,
-0.94100481271744 , 0.67701315879822,
-1.2062630653381 , -0.14108607172966,
-0.95936530828476 , 0.82507961988449,
1.3102972507477 , 4.8358015716076E-02,
-0.95462840795517 , -0.95657634735107,
1.3843868608475 , -4.5906782150269E-02,
-1.0633902549744 , -0.93128252029419,
1.4279329776764 , -0.14106513559818,
1.1497660875320 , 0.882791638374 33,
0.23647691309452 , 1.4383053779602,
-0.81110906600952 , 1.2128539085388,
0.33149531483650 , 1.4158697128296,
-1.2524482011795 , -0.71695524454117,
0.42545720934868 , 1.3614822626114,
-0.60175400972366 , 1.2690199613571,
1.2769542932510 , -0.51767551898956,
0.46760299801826 , -1.2636756896973,
0.60743457078934 , 1.1649851799011,
1.2380973100662 , 0.31721931695938,
0.69398546218872 , 1.0290712118149,
0.15386630594730 , -1.1944670677185,
-0.77655005455017 , -0.87338465452194,
-1.8741229549050E-02 , -1.1353790760040,

APPENDIX C

Copyright @ 1991 AGT Limited

-0.70263749361038 , 0.85432267189026,
-0.19654582440853 , -1.0637385845184,
0.92648190259933 , 0.52192509174347,
0.98265725374222 , -0.37526836991310,
-0.99220246076584 , -0.33656212687492,
0.55053806304932 , 0.89534473419189,
1.0506724119186 , 0.15191125869751,
0.71802806854248 , 0.80500024557114,
1.1011143922806 , -2.6790169999003E-02,
0.87359136343002 , 0.71470791101456,
-0.19459256529808 , -1.1428083181381,
-1.0133891105652 , -0.62734061479568,
-0.34699076414108 , -1.1751155853271,
0.54547435045242 , -1.1340159177780,
0.48006322979927 , 1.1975069046021,
-1.2326081991196 , -0.47131305932999,
0.59059274196625 , 1.2095856666565,
0.40663030743599 , -1.3069387674332,
0.67615872621536 , 1.2111126184464,
1.3554904460907 , 0.35272744297981,
0.73520946502686 , 1.2020268440247,
-1.3775087594986 , -0.31040790677071,
-1.1824613809586 , 0.76710045337677,
-0.27996954321861 , 1.3730301856995,
0.77210420370102 , 1.1527576446533,
1.3428838253021 , 0.26121476292610,
-0.75139236450195 , -1.1134694814682,
1.2886705398560 , 0.25347581505775,
-1.0653619766235 , 0.70698553323746,
-1.2127152681351 , -0.25565749406815,
1.0094038248062 , -0.64167791604996,
1.1179939508438 , 0.26628923416138,
0.94675332307816 , -0.55893445014954,
0.28359553217888 , -1.0080425739288,
-0.46277087926865 , -0.87873333692551,
-0.88684558868408 , -0.30556473135948,
-0.80680304765701 , 0.35761266946793,
0.75870722532272 , 0.33003318309784,
0.24814304709435 , 0.73252141475677,
-0.62811541557312 , -0.35476726293564,
0.65750586986542 , -0.13914376497269,
-0.49959596991539 , -0.37754675745964,
0.58338809013367 , -3.5333048552275E-02,
0.37756574153900 , 0.39624589681625,
0.51176452636719 , 5.8793801814318E-02,
0.40890917181969 , -0.26618802547455,
-0.44414889812469 , -0.13911458849907,
-0.41381889581680 , 0.16923630237579,
-0.38192304968834 , -0.20201873779297,
8.9968837797642E-02 , 0.40955415368080,
-0.24452674388885 , 0.32629245519638,
0.39503705501556 , -3.1020458787680E-02,
0.26439002156258 , -0.27824369072914,

APPENDIX C

Copyright @ 1991 AGT Limited

```
  -0.36956810951233  , -5.6873057037592E-03,
  -0.26016449928284 ,  0.23851068317890,
  -0.33284601569176  , -1.9010026007891E-02,
  -0.23125946521759 ,  0.20754601061344,
  -0.28497639298439  , -8.6228372529149E-03,
   0.17795602977276 , -0.18550240993500,
  -0.22646434605122  ,  2.4967700242996E-02};
double fr1[514];
/*************************************************************************/
main()
{
int     errorre=0,errorim=0,number,mf,init,m,id[257],s[257],i,j,k;
int      jre,jim,l;
double  metricp,metricm,mini1,mini2,jcy;
double  deltphas, xre, xim, curphas[129];

asm(".global declared");
asm ("declared:");

        number = 128;

/************** read the I & Q components & find their amplitude. "fr1" ************/

        for (j=0; j <= number; j++){
        curphas[j] = atan2(z[2*j+1],z[2*j]);
        fr1[2*j] = sqrt(z[2*j]*z[2*j]+z[2*j+1]*z[2*j+1]);}

asm(".global read_data");
asm ("read_data:");

/************ Estimate the Group delay "fr1" from the amplitude "fr1" *************/

        for (k=1; k <= number ;k++){
          fr1[2*k-1] = (fr1[2*k] - fr1[2*k-2])/fr1[2*k];
          fr1[2*k-2] =0.0;}
        fr1[2*k] = 0.0;

/************* estimate the group delay "fr1" using the hilbert transform *************/

      m = 7;

asm(".global b_fft");
asm ("b_fft:");
        fft(number,m,fr1);
asm(".global a_fft");
asm ("a_fft:");

        for(k=1; k<= number/2; k++){
         jcy = fr1[2*k-1]/number;
         fr1[2*k-1] = 0.0-fr1[2*k]/number;
         fr1[2*k]=jcy;}

        for (k=number/2+1; k <= number; k++) {
```

APPENDIX C

Copyright @ 1991 AGT Limited

```
        jcy = fr1[2*k-1]/number;
        fr1[2*k-1]= fr1[2*k]/number;
        fr1[2*k] = 0-jcy;}

asm(".global b_ifft");
asm("b_ifft:");
        ifft(number,m,fr1);
asm(".global a_ifft");
asm("a_ifft:");

/************************** Thresholding & segmenting ************************/

The purpose of this section is to determine the segements of the received frequency frame
where the estimated group delay is larger than a certain threshold. These segments are
labelled

****/

        id[1]=1;
        m = 0;
        init = 0;
        for (k=1; k<=number; k++){
          if(( fr1[2*k-1] > 0.07 )||( fr1[2*k-1] < -0.07)){
            fr1[k] = fr1[2*k-1];
            if(init==0){
              m++;
              id[m] = k;
              init = 1;}}
           else {
            fr1[k] = 0.;
            if(init==1) {
                 m++;
              id[m] = k;
              init = 0;}}
      }
        if(init==1){
          m++;
          id[m] = number + 1;}

  asm(".global threshold");
  asm("threshold:");

  /*************************Group delay Sign Algorithm************************

  The purpose of this part of the software is to determine the proper sign of estimated group
  delay. This is done through defining two Hamming distances one if we assuem the
  estaimted group delay was positive and the other if it is assumed negative and choosing the
  sign that leads to the smaller distance

  ******************************************************************************/

        for (k=1; k<=m; k++){
          metricp = 0.;
```

APPENDIX C

Copyright @ 1991 AGT Limited

```
        metricm = 0.;
        if( (id[k]==id[k+1]-1) || (id[k]==id[k+1]-2) ){
          s[k]=0;
          goto twenty;}

        for (j =id[k]+1; j<=id[k+1]-1; j++){

            deltphas = curphas[j] - curphas[j-1] - fr1[j];

              xrc = 1.4142136377*cos(deltphas);
            xim = 1.4142136377*sin(deltphas);

              mini1 = 1.-xrc;
              mini1 = mini1*mini1;
            if(mini1>((1.+xrc)*(1.+xrc))) mini1=(1.+xrc)*(1.+xrc);
              mini2 = 1.-xim;
              mini2 = mini2*mini2;
            if(mini2>((1.+xim)*(1.+xim))) mini2=(1.+xim)*(1.+xim);

            metricp += mini1 + mini2;

/*****************************************************************************/

            deltphas = curphas[j] - curphas[j-1] + fr1[j];

            xrc = 1.4142136377*cos(deltphas);
            xim = 1.4142136377*sin(deltphas);

              mini1 = 1.-xrc;
              mini1 = mini1 * mini1;
            if(mini1>((1.+xrc)*(1.+xrc))) mini1=(1.+xrc)*(1.+xrc);
              mini2 = 1.-xim;
          mini2 = mini2 * mini2;
            if(mini2>((1.+xim)*(1.+xim))) mini2=(1.+xim)*(1.+xim);

            metricm += mini1 + mini2;

          }
        s[k]=0-1;
        if(metricp < metricm) s[k]=1;
/*      printf("s[%d] = %d\n",k,s[k]);*/
twenty: k++;}

/*****************************************************************************/

      for (k=1; k<=m; k++){
      for (j =id[k]; j<=id[k+1]-1; j++) fr1[j] *= s[k];
/*k++;*/          }
asm(".global a_fs");
asm("a_fs:");

/************************** Decode the Signal **************************/
```

APPENDIX C

Copyright @ 1991 AGT Limited

```
        l = 1;
        while (l < number+1){

          deltphas = curphas[l] - curphas[l-1] - fr1[l];

          xrc = cos(deltphas);
          xim = sin(deltphas);

    jrc = xrc/fabs(xrc);
    jim = xim/fabs(xim);
    if (jrc == jim) {jrc = 0 -jrc;
                jim = 0 -jim;}

            if(l>1){
              if(jrc != jj2[2*l]) errorrc++;

              if(jim != jj2[2*l+1]) errorim++;
                  }         l++;
          }
          printf("%d %d %d\n",i,errorrc,errorim);

    asm(".global end");
    asm("end:");
    }                   /*end of main*/
```

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A transceiver including a transmitter for transmitting electromagnetic signals and a receiver for receiving electromagnetic signals having amplitide and phase differential characteristics, the transmitter comprising:

an encoder for encoding information;

a wideband frequency division multiplexer for multiplexing the encoded information onto wideband frequency channels;

a processor having an output for applying a Fourier transform to the multiplexed information to bring the information into the time domain for transmission;

a low pass filter on the output from the processor;

a local oscillator for upconverting the multiplexed information for transmission; and

further including, in the receiver of the transceiver;

a bandpass filter for filtering the received electromagnetic signals;

a local oscillator for downconverting the received electromagnetic signals to produce output;

a sampler for sampling the output of the local oscillator to produce sampled signals to the channel estimator;

a channel estimator for estimating one or both of the amplitude and the phase differential of the received signals to produce as output one or both of an estimated amplitude and an estimated phase differential respectively; and

B

a decoder for producing signals from the sampled signals and the output from the channel estimator.

2. The transceiver of claim 1 further including, in the receiver of the transceiver:

a deprocessor for applying an inverse Fourier transform to the samples output from the sampler.

3. The transceiver of claim 1, further including, in the receiver of the transceiver:

a power controller before the sampler for monitoring the power of the received signal and for controlling the power of the signal.

4. The transceiver of claim 1, further including, in the transmitter of the transceiver:

a pre-distorter before the processor, the pre-distorter being connected to the channel estimator, for predistorting a signal to be transmitted with one or both of the estimated amplitude or the estimated phase differential.

5. The transceiver of claim 4 in which the power controller is also connected to the pre-distorter for controlling the power of the signal to be transmitted.

6. The transceiver of claim 1 further including:

means to modify the received signal with one or both of the estimated amplitude and phase differential respectively.

B

7. A method for allowing a number of wireless transceivers to exchange frames of information, the method comprising the steps of:

multiplexing a first frame of information over a number of frequencies within a frequency band at a first transceiver to produce multiplexed information;

processing the multiplexed information at the first transceiver;

transmitting the processed information to a second transceiver using a carrier frequency f_c;

receiving the processed information at the second transceiver;

processing the processed information at the second transceiver during a first time interval; and

the width of the frequency band being chosen so that neither carrier nor clock recovery is required at the second transceiver.

8. The method of claim 7 in which the frequency band is formed from a first set of K_1 points and a pair of tail slots each having K_2 points, each of the points being separated by a frequency range Δf, the second transceiver has a maximum expected clock error χT, where T is the duration of one time domain sample, the information is multiplexed over a number M of levels, and K_1 is selected such that $2\pi\chi/K_1 < \pi/M$.

9. The method of claim 8 further including transmitting a second frame of information from the second transceiver to the first transceiver within the same frequency band.

B

10. The method of claim 8 in which K_2 is selected so that the out of band signal is less than a given level.

11. The method of claim 8 in which the first and second transceivers have an expected maximum relative velocity V, the first and second transceivers have carrier frequencies with a frequency offset from each other of f_o, the carrier frequency has a corresponding travelling wavelength λ and Δf is selected so that $[V/(\lambda\Delta f) + f_o/\Delta f]$ is less than or equal to a preselected sampling error.

12. The method of claim 8 in which processing the mutliplexed information at the second transceiver further includes calculating the mean of the phase shift due to sampling error by summing the estimated phase differential of the received signal.

13. The method of claim 12 in which the mean of the phase shift due to sampling error is divided by K_1 and the result removed from the phase differential of the received signal.

14. A transmitter for transmitting electromagnetic signals having amplitide and phase differential characteristics, the transmitter comprising:

an encoder for encoding information;

a wideband frequency division multiplexer for multiplexing the encoded information onto wideband frequency channels;

a processor having an output for applying a Fourier transform to the multiplexed information to

bring the information into the time domain for transmission;

means for operating on the multiplexed information to introduce an arbitrary phase into the multiplexed information;

a low pass filter on the output from the processor; and

a local oscillator for upconverting the multiplexed information for transmission.

15. The transmitter of claim 14 further in combination with a receiver for receiving electromagnetic signals, the receiver comprising:

a bandpass filter for filtering the received electromagnetic signals;

a local oscillator for downconverting the received electromagnetic signals to produce output;

a sampler for sampling the output of the local oscillator to produce sampled signals to the channel estimator;

a channel estimator for estimating one or both of the amplitude and the phase differential of the received signals to produce as output one or both of an estimated amplitude and an estimated phase differential respectively; and

a decoder for producing signals from the sampled signals and the output from the channel estimator.

16. The transmitter and receiver combination of claim 15 further including, in the receiver:

a deprocessor for applying an inverse Fourier transform to the samples output from the sampler.

B

17. The transmitter and receiver combination of claim 16, further including, in the receiver:

a power controller before the sampler for monitoring the power of the received signal and for controlling the power of the signal.

18. The transmitter and receiver combination of claim 15, further comprising:

a pre-distorter before the processor, the pre-distorter being connected to the channel estimator, for predistorting a signal to be transmitted with one or both of the estimated amplitude or the estimated phase differential.

19. The transcmitter and receiver combination of claim 18 in which the power controller is also connected to the pre-distorter for controlling the power of the signal to be transmitted.

20. The transmitter and receiver combination of claim 15 further including:

means to modify the received signal with one or both of the estimated amplitude and phase differential respectively.

21. A method for allowing a number of wireless transceivers to exchange frames of information symbols, the method comprising the steps of:

multiplexing a first frame of information symbols over a number of frequencies within a frequency band at a first transceiver to produce multiplexed information;

processing the multiplexed information at the first transceiver, including introducing an arbitrary phase into the information symbols;

B

transmitting the processed information to a second transceiver using a carrier frequency f_c;

receiving the processed information at the second transceiver; and

processing the processed information at the second transceiver during a first time interval.

22. The method of claim 21 in which the width of the frequency band being chosen so that neither carrier nor clock recovery is required at the second transceiver.

23. The method of claim 22 in which the frequency band is formed from a first set of K_1 points and a pair of tail slots each having K_2 points, each of the points being separated by a frequency range Δf, the second transceiver has a maximum expected clock error χT, where T is the duration of one time domain sample, the information is multiplexed over a number M of levels, and K_1 is selected such that $2\pi\chi/K_1 < \pi/M$.

24. The method of claim 23 further including transmitting a second frame of information from the second transceiver to the first transceiver within the same frequency band.

25. The method of claim 23 in which K_2 is selected so that the out of band signal is less than a given level.

26. The method of claim 23 in which the first and second transceivers have an expected maximum relative velocity V, the first and second transceivers have carrier frequencies with a frequency offset from

each other of f_o, the carrier frequency has a corresponding travelling wavelength λ and Δf is selected so that $[V/(\lambda\Delta f) + f_o/\Delta f]$ is less than or equal to a preselected sampling error.

27. The method of claim 23 in which processing the mutliplexed information at the second transceiver further includes calculating the mean of the phase shift due to sampling error by summing the estimated phase differential of the received signal.

28. The method of claim 27 in which the mean of the phase shift due to sampling error is divided by K_1 and the result removed from the phase differential of the received signal.

29. A transmitter for transmitting electromagnetic signals having amplitide and phase differential characteristics, the transmitter comprising:

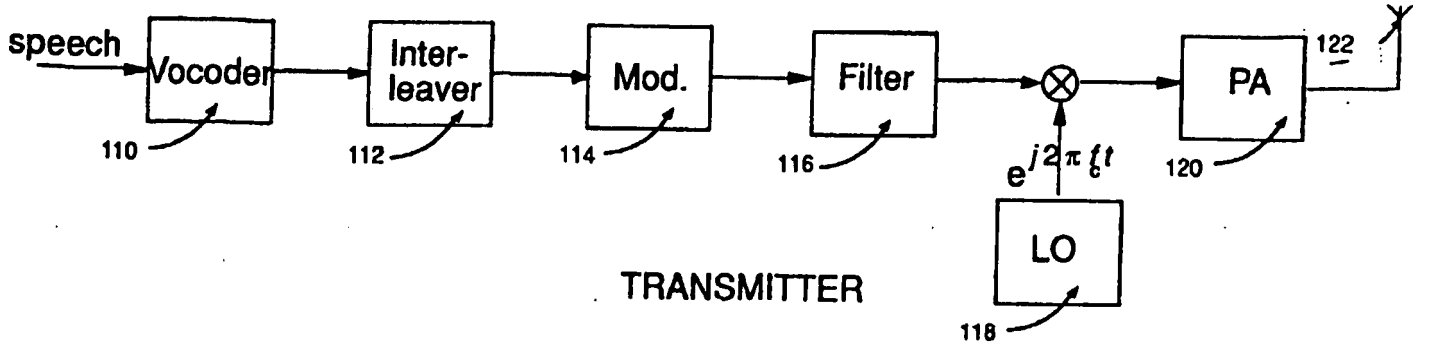
an encoder for encoding information;

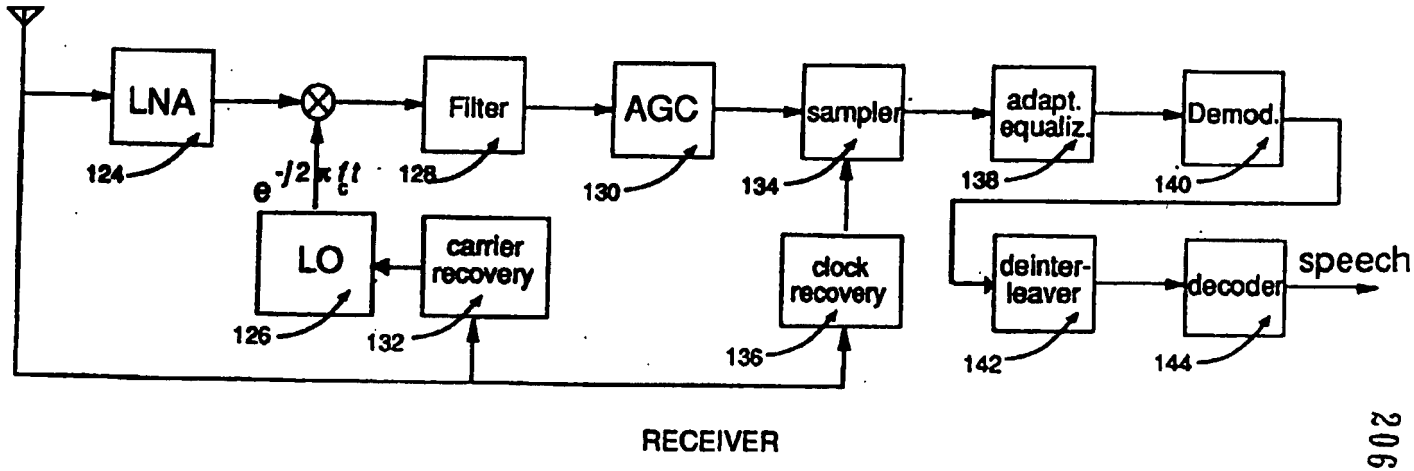
a wideband frequency division multiplexer for multiplexing the encoded information onto wideband frequency channels;

a processor having an output for applying a Fourier transform to the multiplexed information to bring the information into the time domain for transmission;

a low pass filter on the output from the processor; and

a local oscillator for upconverting the multiplexed information for transmission.





2064975

Fig. 1

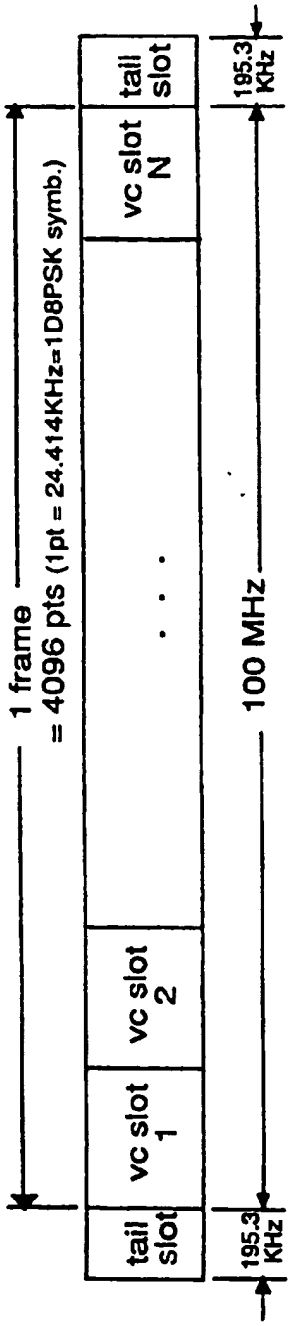

1 frame
= 4096 pts (1pt = 24.414KHz=1D8PSK symb.)
tail slot
vc slot 1
vc slot 2
. . .
vc slot N
tail slot
195.3 KHz
100 MHz
195.3 KHz

Fig. 2

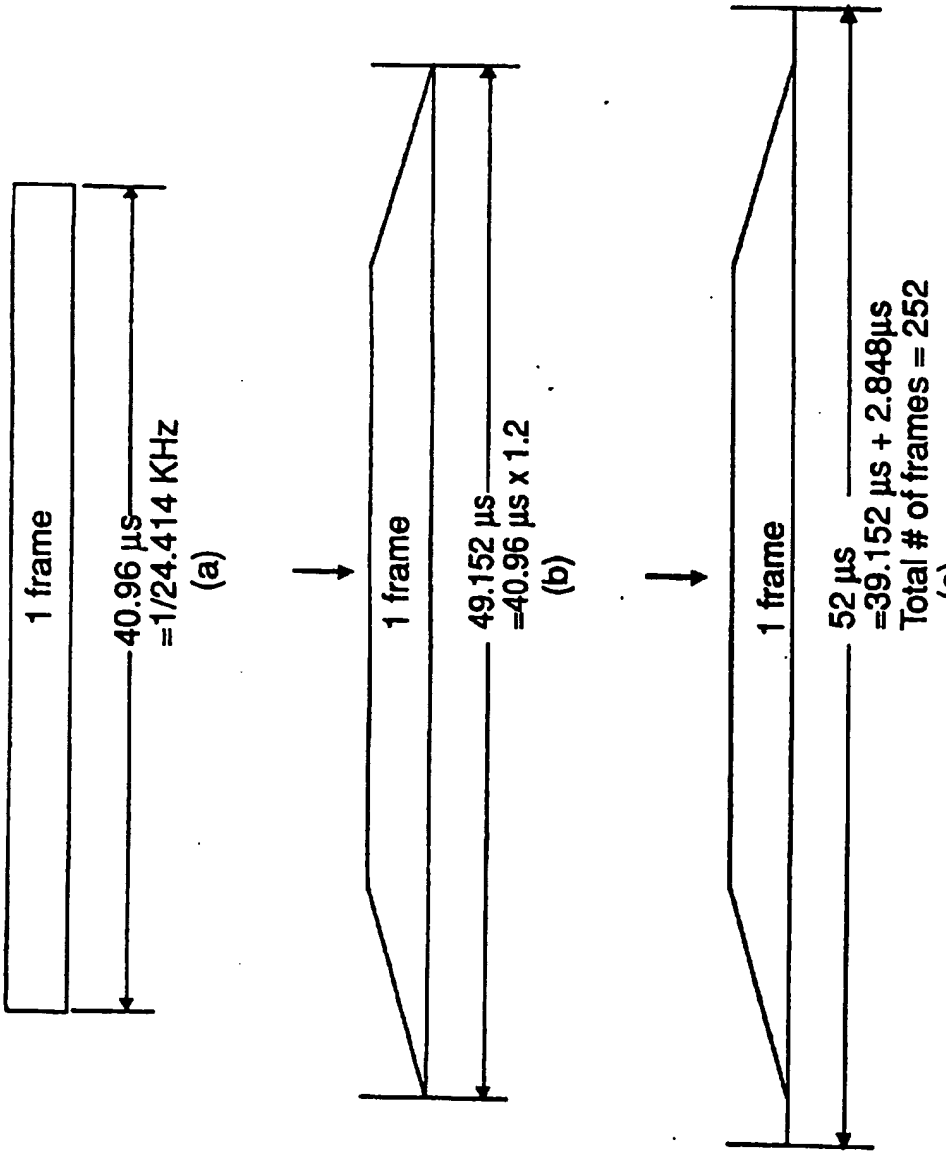

1 frame
40.96 µs
=1/24.414 KHz
(a)
1 frame
49.152 µs
=40.96 µs x 1.2
(b)
1 frame
52 µs
=39.152 µs + 2.848µs
Total # of frames = 252
(c)

Fig . 3

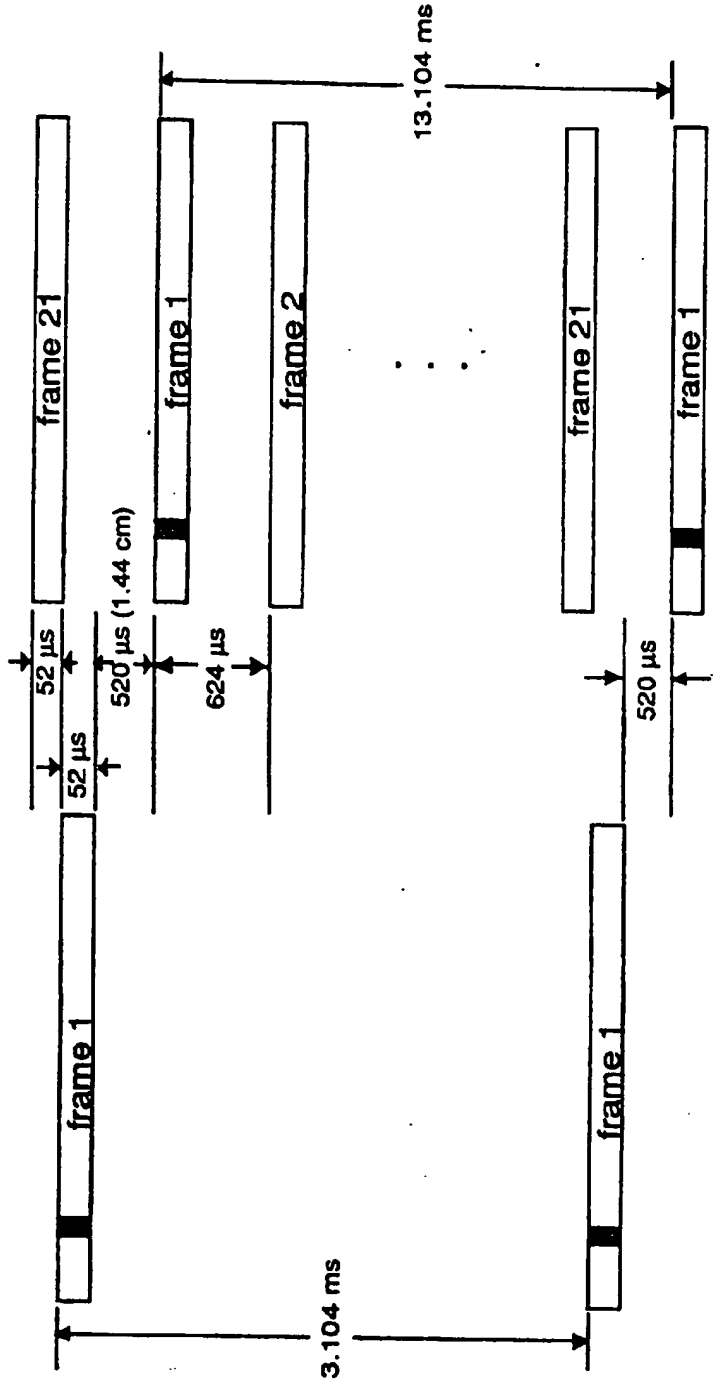
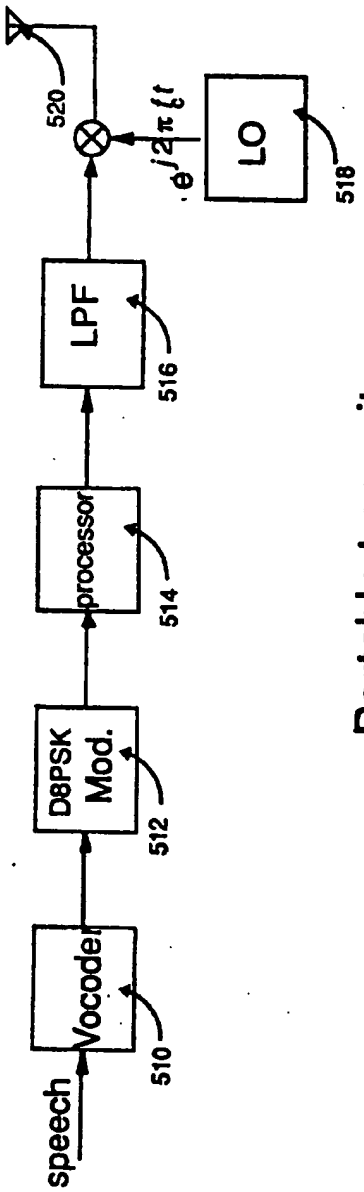

frame 21
frame 1
frame 2
frame 21
frame 1
13.104 ms
52 µs
52 µs
520 µs (1.44 cm)
624 µs
520 µs
frame 1
frame 1
13.104 ms
BS transmits
portable transmits

Fig. 4



Portable transmits

Fig. 5a

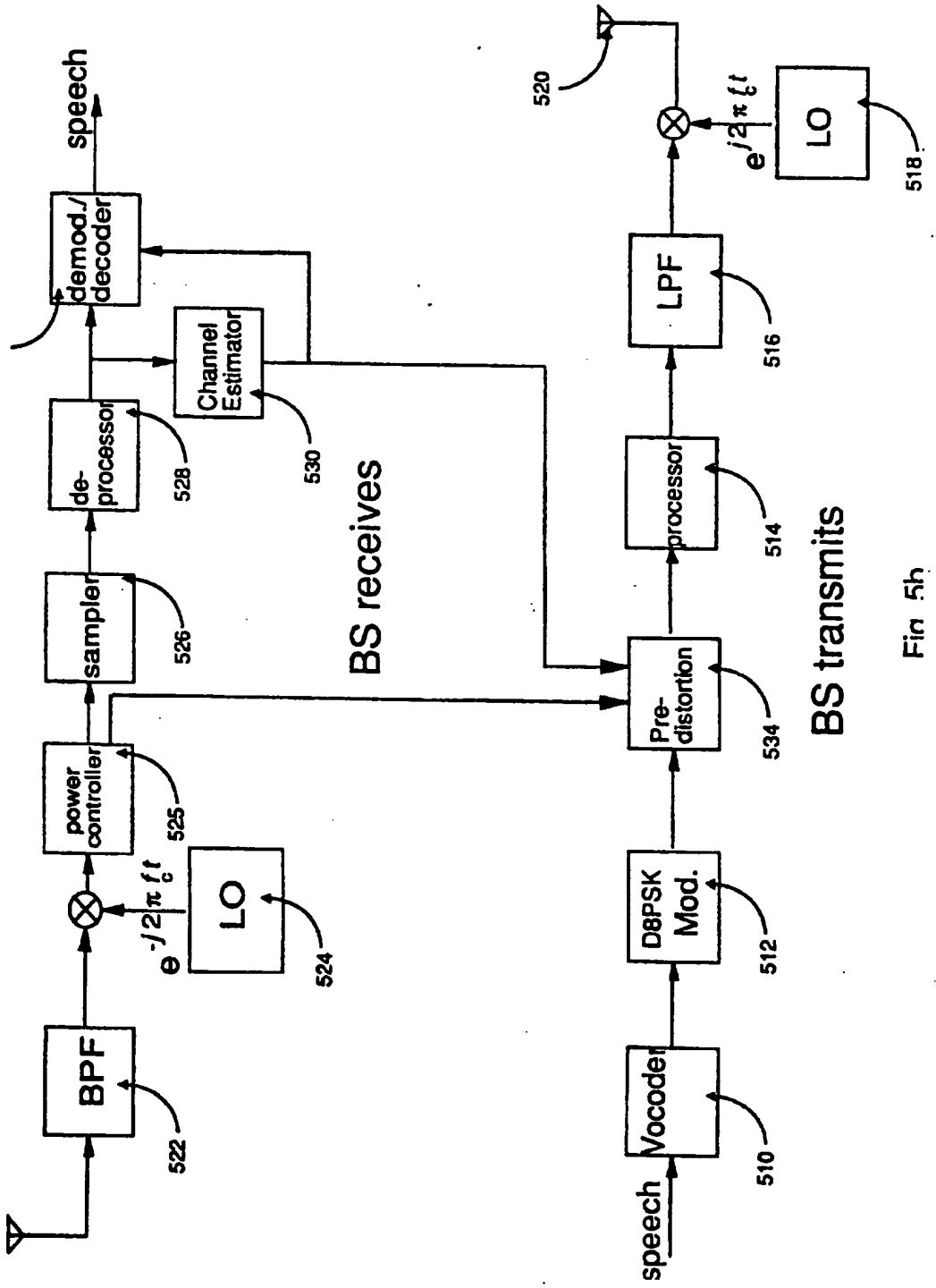
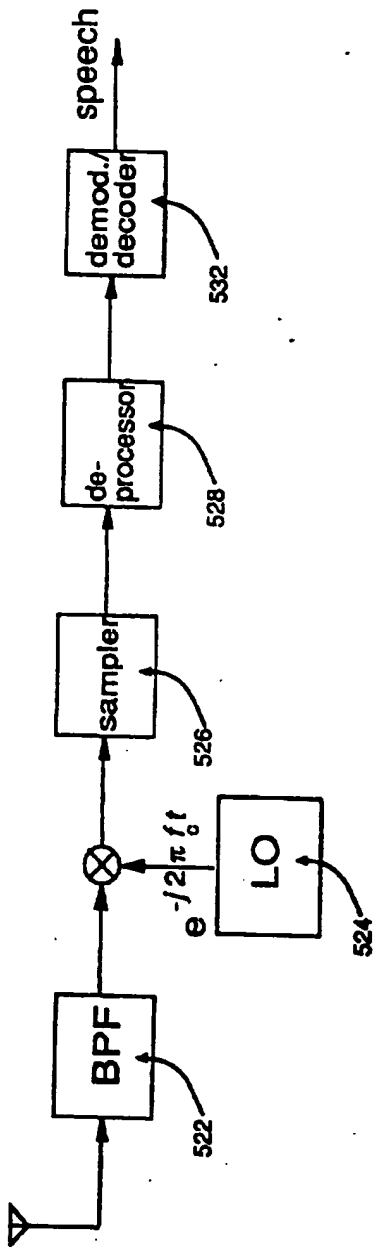
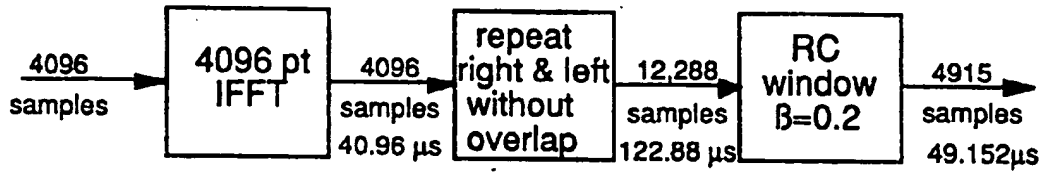

speech
demod./ decoder
532
Channel Estimator
530
de-processor
528
sampler
526
power controller
525
BPF
522
LO
524
$e^{-j2\pi f_c t}$
BS receives
520
LO
518
$e^{j2\pi f_c t}$
LPF
516
processor
514
Pre-distortion
534
D8PSK Mod.
512
Vocoder
510
speech
BS transmits

Fig 5b



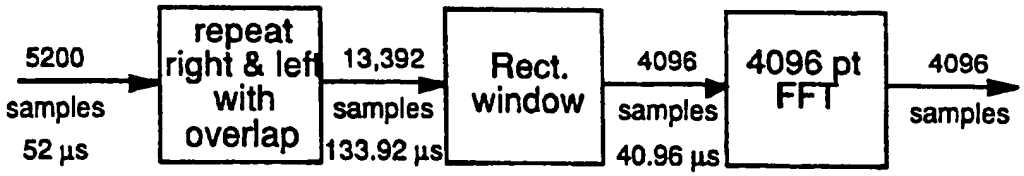
Portable receives

Fig. 5c



PROCESSOR

(a)

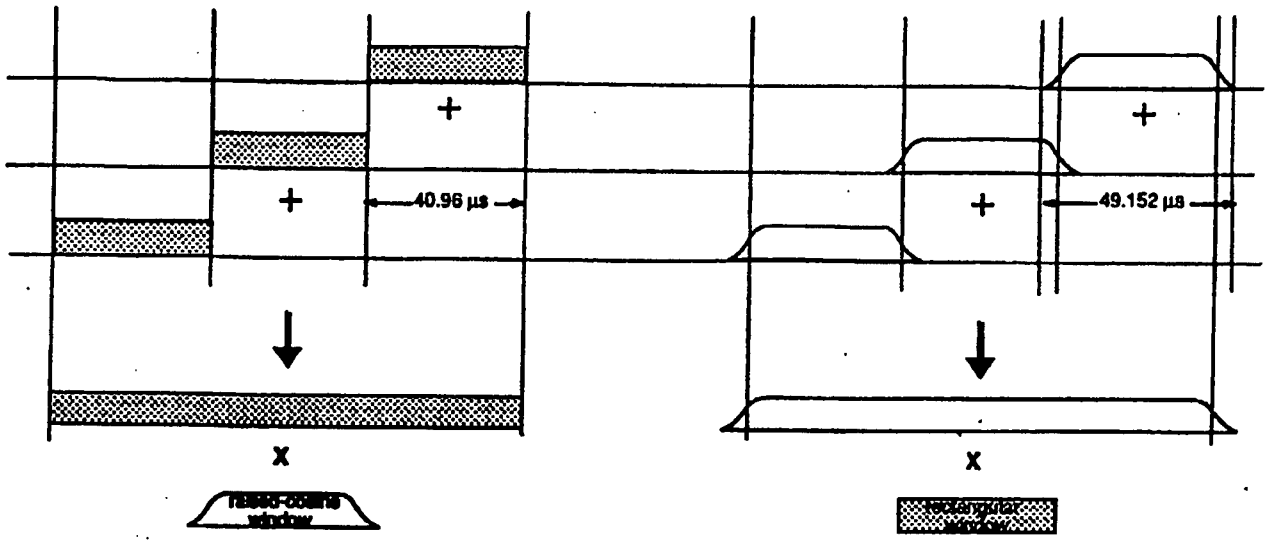


DE-PROCESSOR

(b)

† a sample above is a complex sample.

Fig. 6a & b



Repeat right & Left without overlap followed by a raised cosine window

(last 2 blocks in processor)

Repeat right & left with overlap followed by a rectangular window

(last 2 blocks in de-processor)

Fig. 6c

2064975

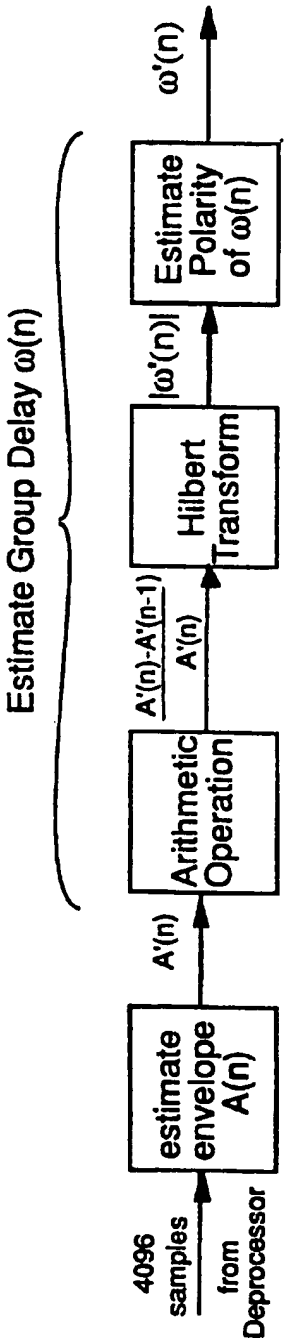
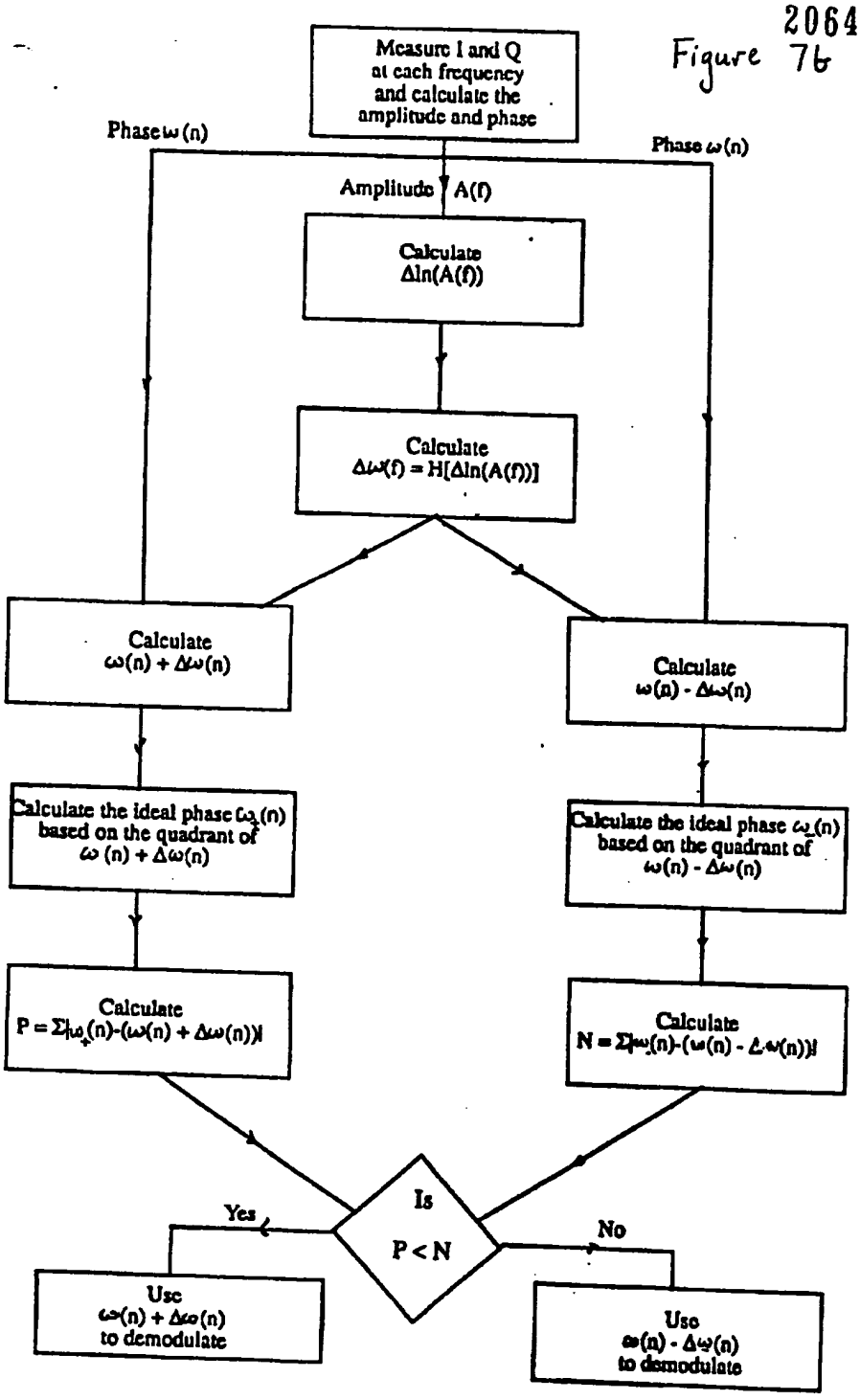
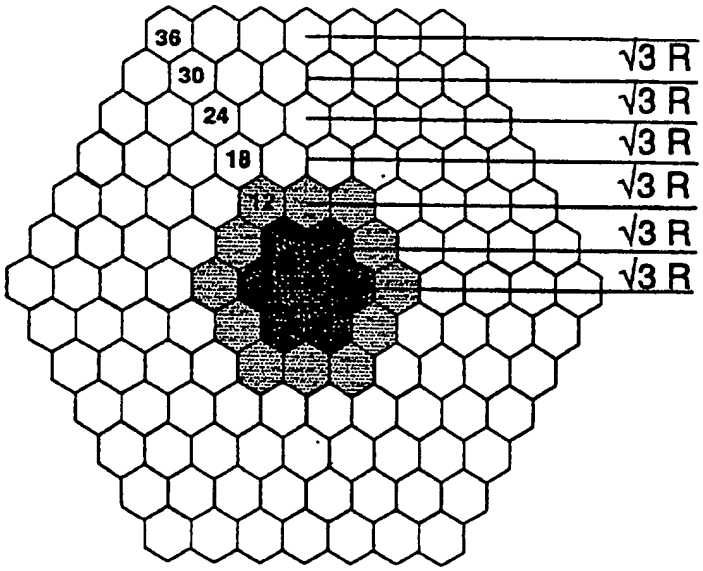

4096 samples from Deprocessor
estimate envelope A(n)
A'(n)
Arithmetic Operation
A'(n)-A'(n-1) / A'(n)
Hilbert Transform
|ω'(n)|
Estimate Polarity of ω(n)
ω'(n)
Estimate Group Delay ω(n)

fig. 7a

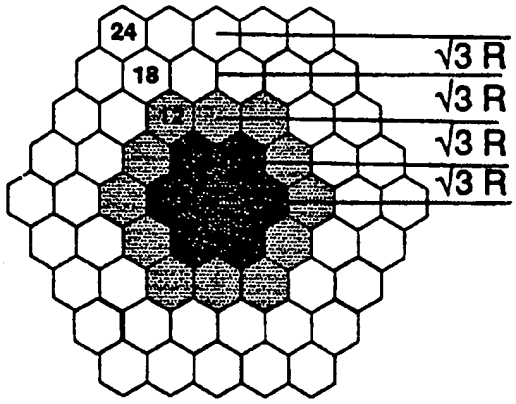
2064975

Figure 7b

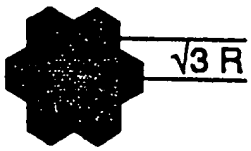




(a)



(b)



(c)

Fig. 8

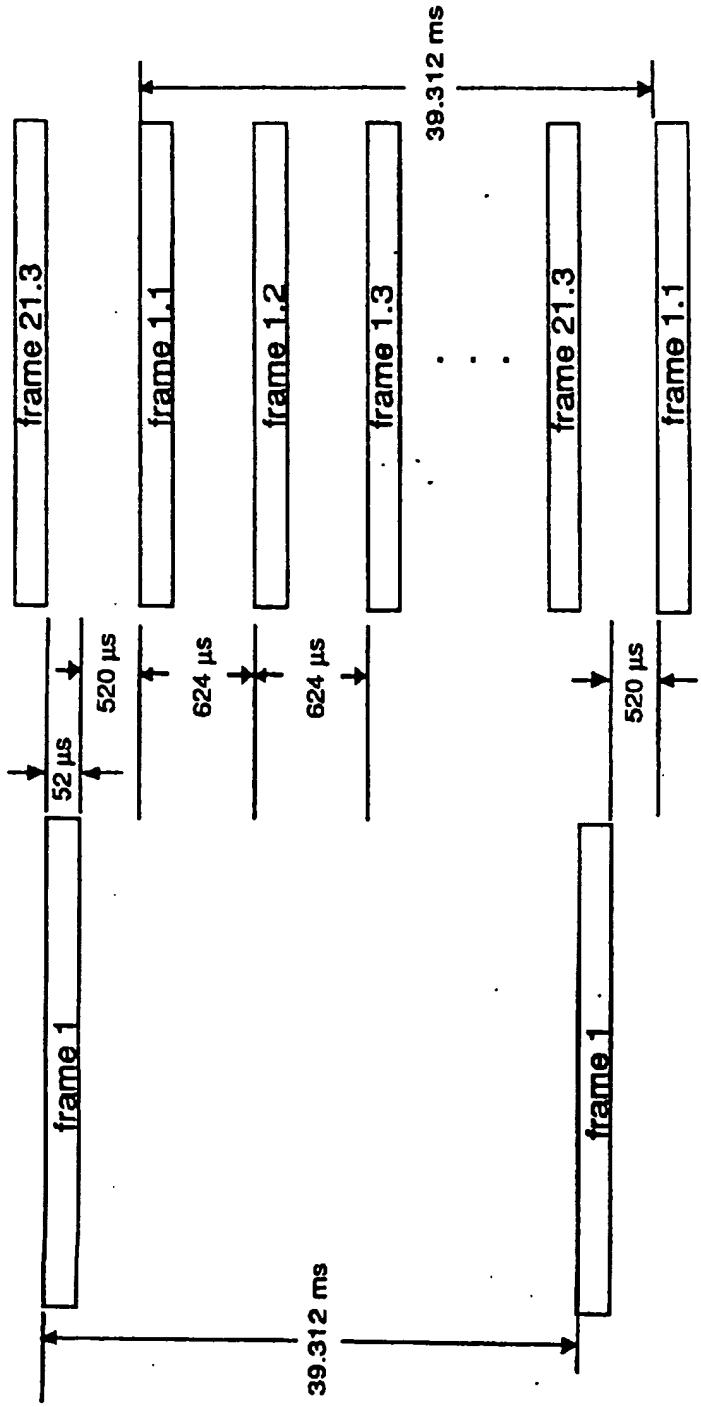

frame 21.3
frame 1.1
frame 1.2
frame 1.3
frame 21.3
frame 1.1
39.312 ms
52 µs
520 µs
624 µs
624 µs
520 µs
frame 1
frame 1
39.312 ms
Portable Transmits once.
BS Transmits thrice.

Fig. 9a

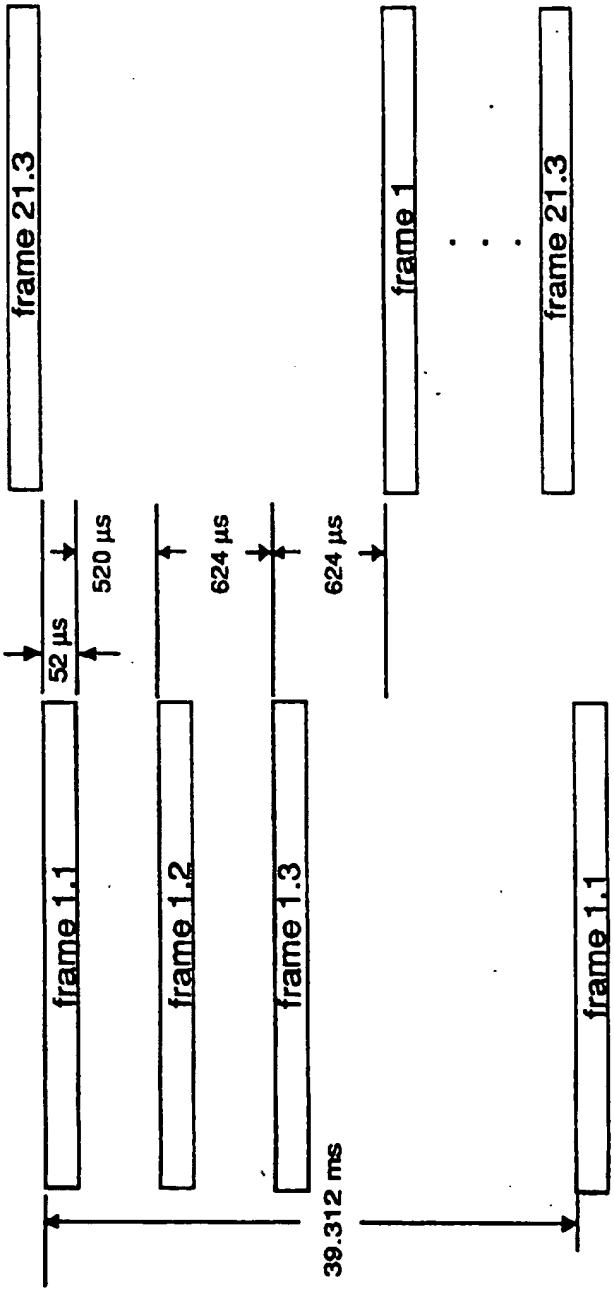

frame 21.3
frame 1.1
frame 1.2
frame 1.3
frame 1
frame 21.3
frame 1.1
52 µs
520 µs
624 µs
624 µs
39.312 ms
Portable Transmits thrice.
BS Transmits once.

Fig. 9b

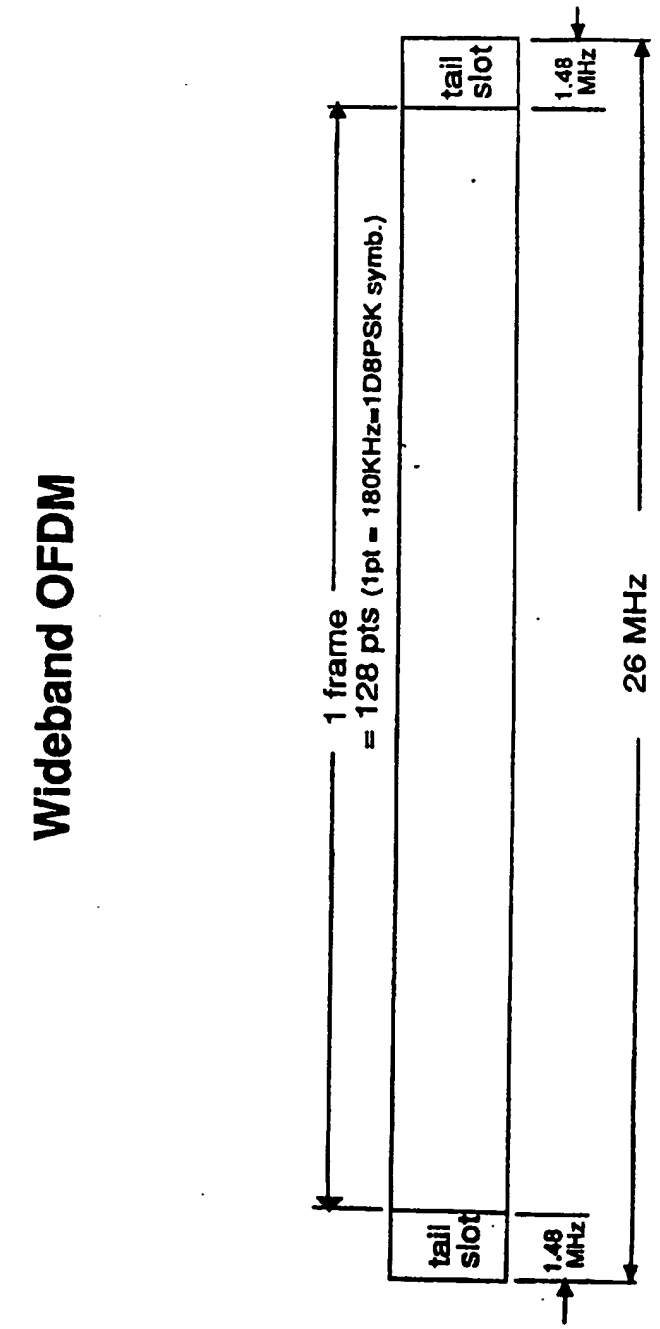

Wideband OFDM
1 frame
= 128 pts (1pt = 180KHz=1D8PSK symb.)
tail slot
tail slot
1.48 MHz
1.48 MHz
26 MHz

Fig. 10

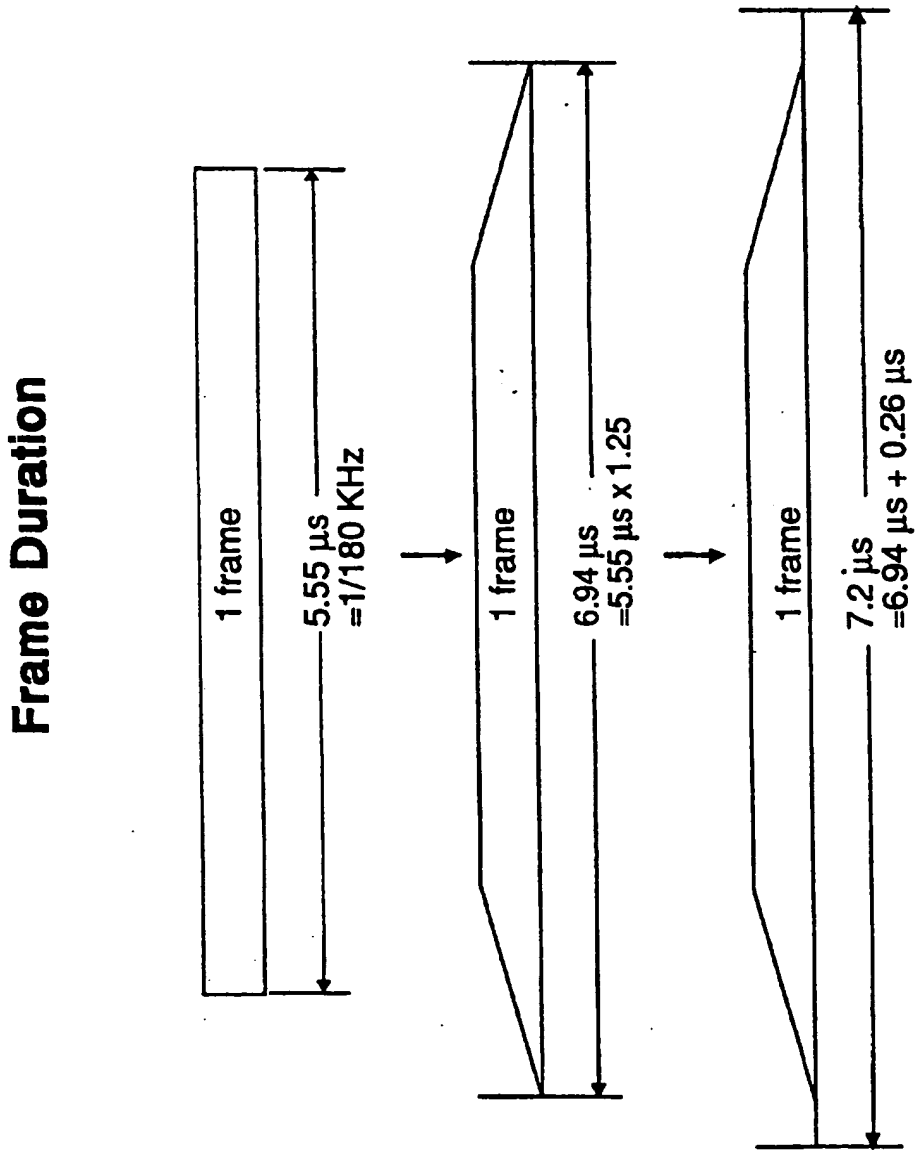

Frame Duration
1 frame
5.55 µs
=1/180 KHz
1 frame
6.94 µs
=5.55 µs x 1.25
1 frame
7.2 µs
=6.94 µs + 0.26 µs

Fig. 11

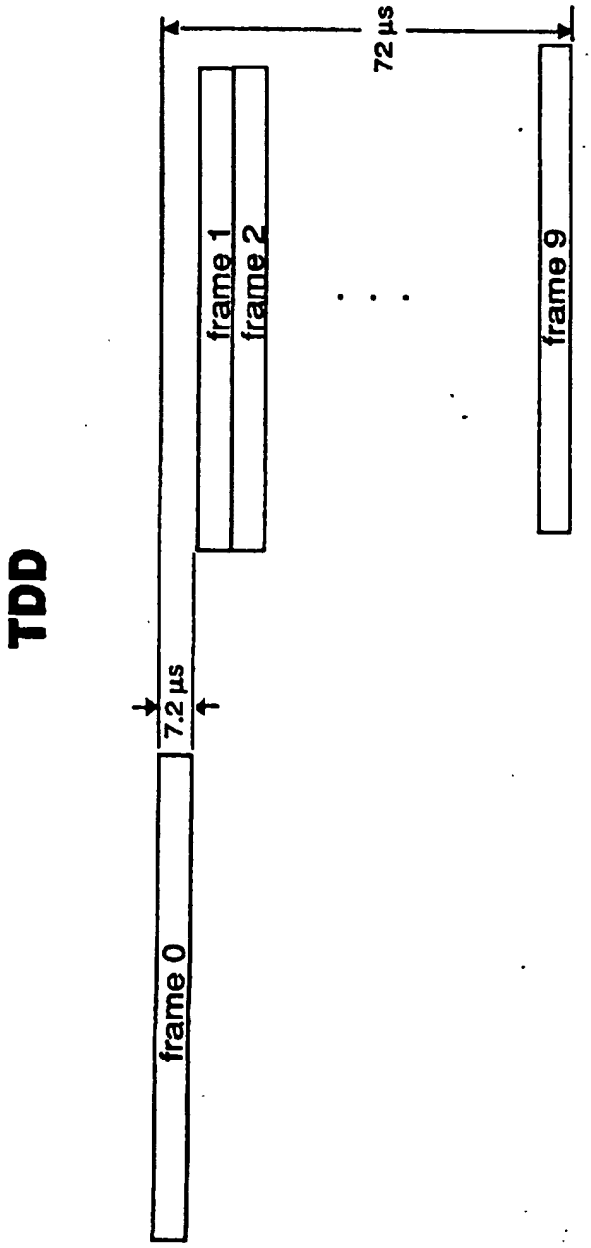
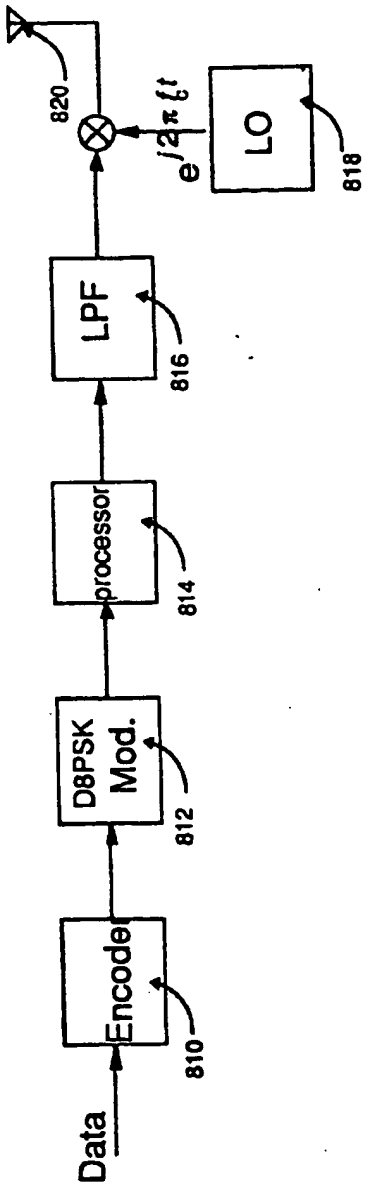

TDD
72 µs
frame 1
frame 2
frame 9
7.2 µs
frame 0
Second Transceiver transmits
First Transceiver transmits

Fig. 12



First Transceiver transmits

Fig. 13a

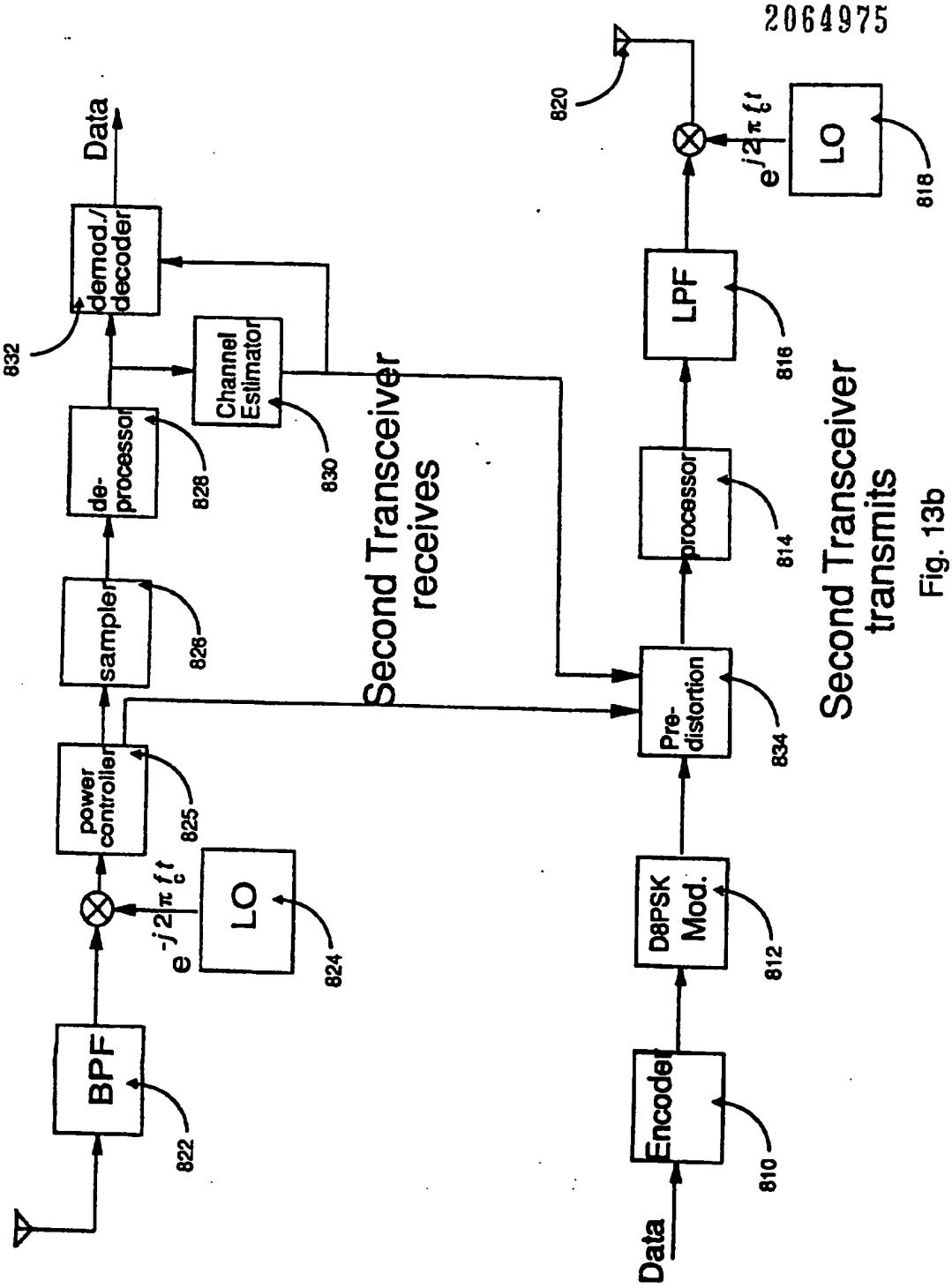
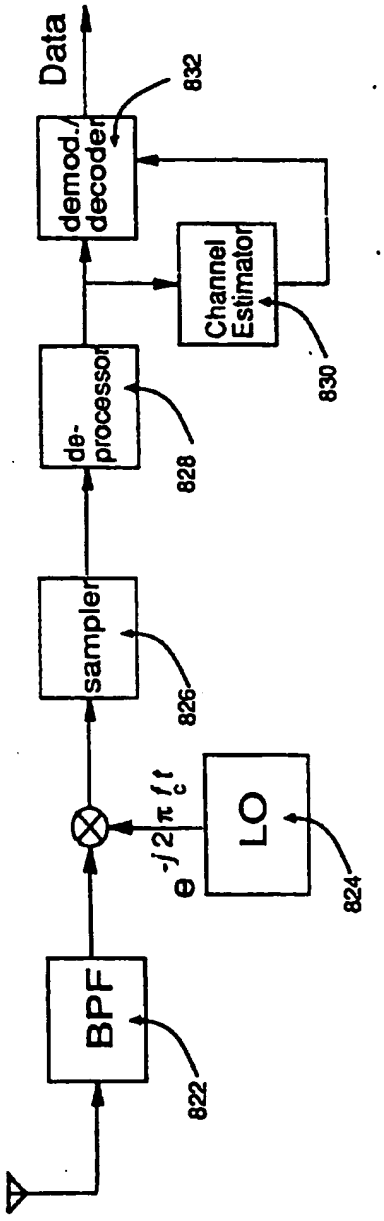

Data
demod./ decoder
832
Channel Estimator
830
de-processor
828
sampler
826
power controller
825
BPF
822
LO
824
$e^{-j2\pi f_c t}$
Second Transceiver receives
Data
Encoder
810
D8PSK Mod.
812
Pre-distortion
834
processor
814
LPF
816
LO
818
$e^{j2\pi f_c t}$
820
Second Transceiver transmits

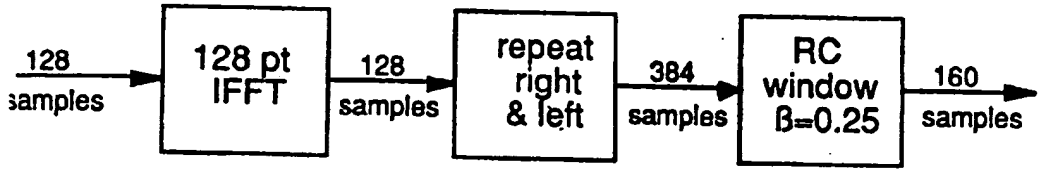
Fig. 13b



First transceiver receives

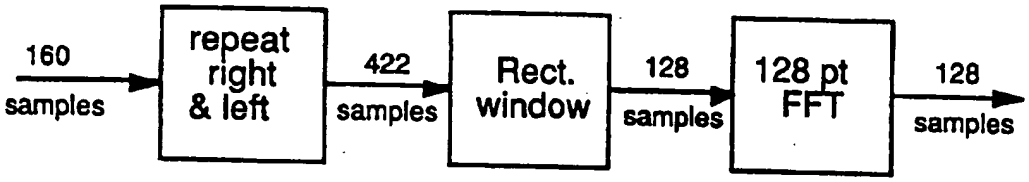
Fig. 13c

Digital Signal Processors



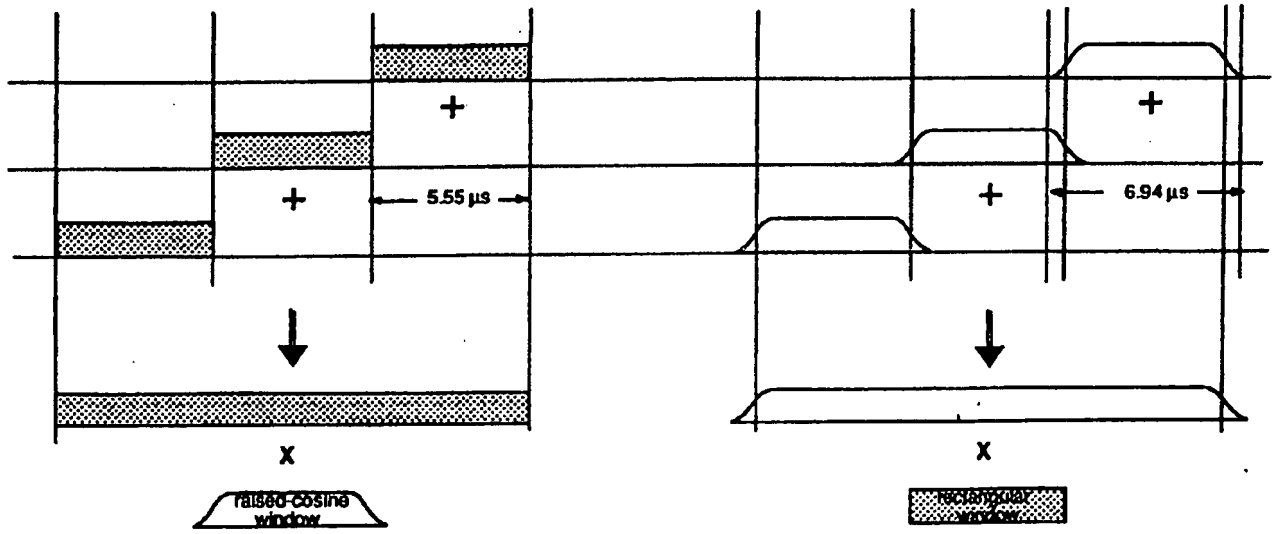
PROCESSOR

Fig. 14a



DE-PROCESSOR

Fig. 14b



Repeat right & Left without overlap followed by a raised cosine window

(last 2 blocks in processor)

Repeat right & left with overlap followed by a rectangular window

(last 2 blocks in de-processor)

Fig. 14c

2064975

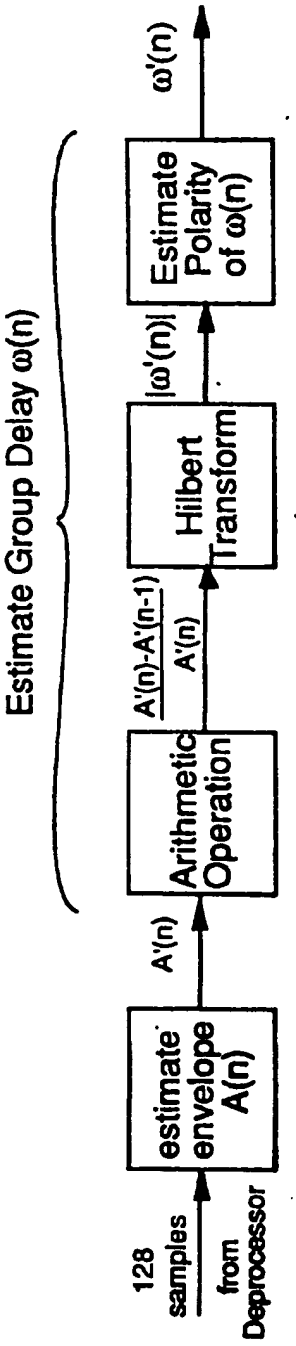

Estimate Group Delay ω(n)
128 samples from Deprocessor
estimate envelope A(n)
A'(n)
Arithmetic Operation
A'(n)-A'(n-1) / A'(n)
Hilbert Transform
|ω'(n)|
Estimate Polarity of ω(n)
ω'(n)

Fig. 15

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.